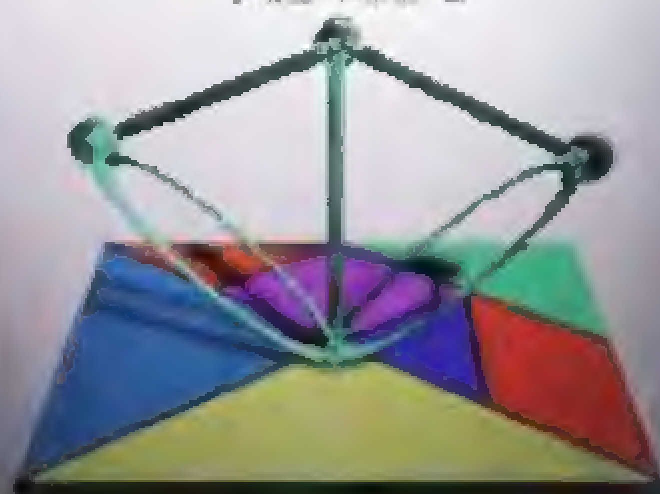
图论及其应用

（第2版）



(京)新登字 158 号

内 容 提 要

图论研究的问题有的源远流长，可追溯到欧拉。它成为数学一活跃分支则是近 30 年的事。60 年代以来发现它在许多领域有着广泛的应用，特别是计算机科学、电路网络等，图论的引进改变了它们的面貌。

本书以讲述图论的应用为主，介绍它解决问题的思想和算法。全书分基本理论篇和应用篇两大部分，其中有些是很新很热门的课题。

本书可作为大学数学专业、计算系理论专业的教材。相关学科的科技工作者也可从中找到他们所需要的材料。

图书在版编目(CIP)数据

图论及其应用/卢开澄，卢华明著. —2 版. —北京：清华大学出版社，1995
ISBN 7-302-01817-0

Ⅰ.图… Ⅱ.①卢… ②卢… Ⅲ.图论 Ⅳ.O157.5

中国版本图书馆 CIP 数据核字(95)第 03770 号

出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)
http://www.tup.tsinghua.edu.cn
责任编辑：杨 靖
印 刷 者：清华大学印刷厂
发 行 者：新华书店总店北京发行所
开 本：787×1092 1/16 印张：14.5 字数：334 千字
版 次：1995 年 8 月第 2 版 2002 年 12 月第 5 次印刷
书 号：ISBN 7-302-01817-0/TP・813
印 数：11001～12500
定 价：12.80 元

计算机科学组合学丛书

前　　言

电子计算机的出现是20世纪的大事，它改变了我们这个世界的面貌。可以毫不夸张地说，它的影响遍及所有的角落，几乎无处不感觉到它的存在。数学更不例外。严格地说，电子计算机本身就是近代数学的辉煌成就。将计算机与数学割裂开来，既不合理也不可能。组合学也就是在计算机科学蓬勃发展的刺激下而崛起的，从而成为近若干年来最活跃的数学分支。它研究的问题有的可追溯到欧拉和哈密尔顿等18世纪的数学家，但它成为一新的分支还是近若干年的事。它从与计算机科学相结合中获得了广阔的发展空间，从而也为计算机科学奠定了理论基础。

什么是计算机科学呢？有的学者将它定义为研究算法的一门学科。研究算法无疑是计算机科学的重要领域，也是本丛书的核心内容，贯穿始终。组合学家在20世纪70年代初建立的算法复杂性"NP理论"，至今仍然令无数计算机科学工作者与数学工作者为之折腰。

计算机科学里的组合学内容十分广泛。本丛书涉及组合分析、图论、组合算法、近代密码学、编码理论及算法复杂性等七部分。

组合分析是算法的理论基础。组合分析之与组合算法犹如数学分析之与计算数学，众所周知，前者是后者的理论根基。

图论原本是组合数学这个"家族"的主要成员，只因它已成长壮大，故自立门户独立出去。

算法复杂性的NP理论是近30年的一大成就。研究表明对于一类叫做NPC类的困难问题，至今都不存在有效算法，但它们难度相当，只要其中任何一个找到多项式解法，则全体都获得解决；或证明它们根本不存在有效办法。不论是前者还是后者都还看不见露到海平面上的桅杆塔、它吸引了众多的有志之士。密码学是其中十分引人入胜的分支。如若设计好的密码，对它的破译等价于某一NPC类困难问题，无疑这样的密码将是牢不可破的。

在计算机网络深入普及的信息时代，信息本身就是时间，就是财富。信息的传输通过的是脆弱的公共信道，信息储存于"不设防"的计算机系统中，如何保护信息的安全使之不被窃取及不至于被篡改或破坏，已成为当今被普遍关注的重大问题。密码是有效而且可行的办法。在计算机网络的刺激下，近代密码学便在算法复杂性理论的基础上建立起来了。密码作为一种技术，自从人类有了战争，不久便有了它。但作为一门学科则是近20多年的事。甚至于它已成为其它学科的基础。密码也从此走出"军营"，进入百姓家。

实际中的"优化"问题是大量的，半个多世纪以来它曾经几度辉煌。近来在计算机科学的影响下，又出现了若干闪光点，十分耀眼，引人注目。

实际上密码也是一种编码。如果说密码学研究的编码是保证通信的保密与安全，则编码理论研究的是通信中如何纠错与检错。计算机纠错码是既实用、理论上又饶有趣味的分支。

本丛书是作者在清华大学计算机科学与技术系长期工作的总结。它不是一部"长篇"记述，而是互相关联又彼此相对独立，因此难免有少量交叉。它们涉及的面如此之广，囿于作者的水平，缺点和错误在所难免，敬请读者不吝指正。谢谢。

第 2 版 序

图论是一门既古老而又年轻的学科。说它古老，因为它可以追溯到 17 世纪的 Euler。讲图论没有不提到 Könisberg 桥的问题，Euler 解决它用到图的方法确是非常典型的例子。但它成为一门学科，还是近 30 年的事。

近若干年来，在计算机科学蓬勃发展的刺激下，图论也获得一个很大的空间。在计算机的许多领域里，它都占有一席之地，有了自己的位置。不仅如此，在物理学、生物学、电力工程、运筹学、以及社会科学等领域都有它的应用。可以这么说，图论之所以成为图论，是因为它显示了很好的应用前景。

本书从第 1 版出版到现在，已超过 10 年。作者根据这几年的实践，对它作了比较彻底的改写。全书共七章，分基础理论篇和应用篇两部分。前一部分及第五章由卢华明执笔，增加了许多新内容，比如 Petri 网，它是近若干年新兴起很有前途的分支。对研究并行计算、复杂系统等有帮助。图论作为离散数学的成员没有理由将 Petri 网拒之门外。

本书仍以研究算法为主，以学以致用为目的。当然错误和缺点在所难免，望读者不吝指教。

作　者

1994 年 10 月

目　录

第一部分　基础理论篇

第二部分　应　用　篇

第一章　图的基本概念

§1　引　　论

图论是一新的数学分支，也是一门很有实用价值的学科，它在自然科学、社会科学等各领域均有很多应用。近年来它受计算机科学蓬勃发展的刺激，发展极其迅速。应用范围不断拓广，已渗透到诸如语言学、逻辑学、物理学、化学、电讯工程、计算机科学以及数学的其它分支中。特别在计算机科学中，如形式语言、数据结构、分布式系统、操作系统等方面均扮演着重要的角色。

首先我们来看几个实际的问题。

一、Könisberg 七桥问题（或 Euler 回路问题）

图论所研究的问题源远流长。远在 18 世纪，著名的 Könisberg 七桥问题就是当时很有趣的问题。Könisberg 城在 18 世纪属于东普鲁士，它位于 Pregel 河畔，河中有两个小岛，河两岸和河中两岛，通过七座桥彼此相连（见图 1-1-1）。

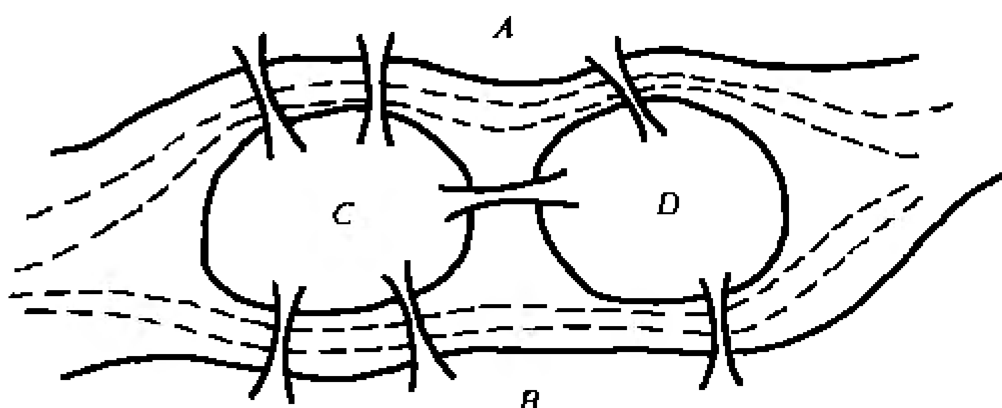


图　1-1-1

有一趣味问题：游人从两岸 A，B 或两个小岛 C，D 中任一个地方出发，要找到一条路线做到每座桥恰通过一次而最后返回原地。问题看来不复杂，但谁也解决不了，也说不出其所以然来。1736 年，当时著名的数学家 Euler 仔细研究了这个问题，他将上述四块陆地与七座桥间的关系用一个抽象图形来描述（见图 1-1-2），其中 A、B、C、D 分别用四个点来表示，而陆地之间有桥相连者则用连接两个点的连线来表示，这样上述的 Könisberg 七桥问题就变成为由点和边所组成的图 1-1-2 的

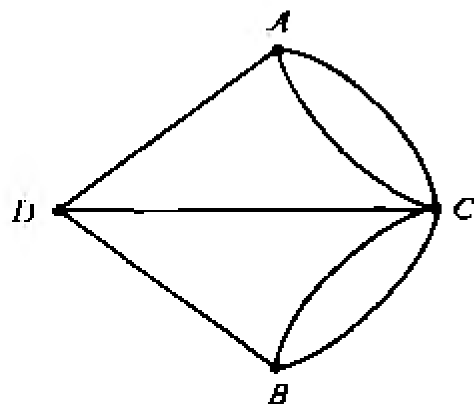


图　1-1-2

如下问题：

试求从图中的任一点出发，通过每条边一次，最后返回到该点，这样的路径是否存在？于是问题就变得简洁明了多了，同时也更一般、更深刻。这就是图论中的 Euler 问题。

关于 Könisberg 七桥问题的回答是否定的。直观上不难发现，为了要回到原来的地方，要求与每一个顶点相关联的边的数目，均应为偶数，从而可得从一条边进入，而从另一条边出去，一进一出才行。在此基础上，Euler 找到了一般的图存在这样一条回路的充分而且必要条件，详细讨论见本章第二节。

二、路径问题

如图 1-1-3 所示。顶点 $v_1, v_2, \cdots, v_7$ 代表七座城市，有方向的边$\overrightarrow{v_iv_j}$表示从 v_i 城到 v_j 城的单行车道，问从 v_1 城到 v_7 城有无道路相通？

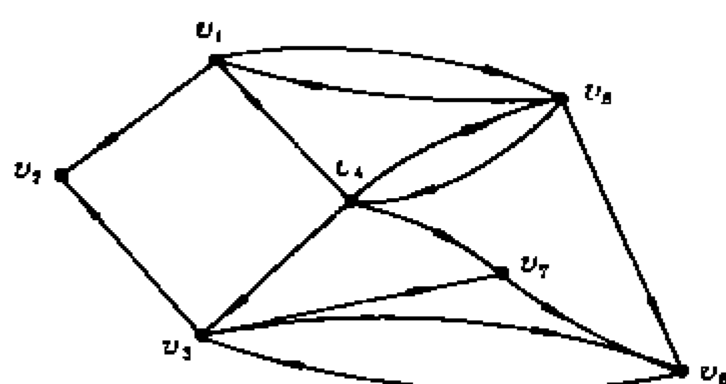


图 1-1-3

这个例子很简单，观察图 1-1-3 就不难给出解答。如果我们进一步问：若 v_1 城到 v_7 城有道路相通，共有几条不同的道路？每条道路中间经过哪几个城市？一般的这类问题就不能简单地通过观察法来解决。下面讨论的是路径问题的一般算法。

为此引进矩阵

$$A = (a_{ij})_{7\times 7},$$

其中

$$a_{ij} = \begin{cases} 1，若从 v_i 点到 v_j 点有边(v_i, v_j) 相连； \\ 0，若从 v_i 点到 v_j 点无边相连。 \end{cases}$$

这样我们得到

$$A = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{matrix} \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$
$$\quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6 \quad v_7$$

而且得

$$A^2=A\cdot A=\begin{pmatrix}1&0&0&1&0&1&0\\0&0&0&0&0&0&0\\0&0&1&0&0&1&0\\1&2&0&1&1&3&1\\1&1&2&0&2&0&1\\0&1&0&0&0&1&1\\0&0&1&0&0&0&0\end{pmatrix}=(a_{ij}^{(2)}),$$

其中

$$a_{ij}^{(2)}=\sum_{k=1}^{7}a_{ik}a_{kj}\text{。}$$

同样可得到

$$A^3=A^2\cdot A=A\cdot A^2=(a_{ij}^{(3)})=\begin{pmatrix}1&1&2&0&2&0&1\\0&0&0&0&0&0&0\\0&1&1&0&0&1&1\\2&1&4&1&2&2&1\\2&3&0&2&1&5&2\\0&0&1&0&0&1&0\\0&1&0&0&0&1&1\end{pmatrix},$$

其中

$$a_{ij}^{(3)}=\sum_{k=1}^{7}a_{ik}^{(2)}a_{kj}=\sum_{k=1}^{7}a_{ik}a_{kj}^{(2)}\text{。}$$

一般有：

$$A^k=(a_{ij}^{(k)})_{7\times 7},$$

其中

$$a_{ij}^{(k)}=\sum_{k=1}^{7}a_{ik}^{(k-1)}a_{kj}\text{。}$$

现在来看看 $a_{ij}^{(k)}$ 的值具有什么实际意义。以 $a_{ij}^{(2)}$ 为例：

$$a_{ij}^{(2)}=\sum_{k=1}^{7}a_{ik}a_{kj}=a_{i1}a_{1j}+a_{i2}a_{2j}+\cdots+a_{i7}a_{7j},$$

$a_{ik}\cdot a_{kj}\neq 0$ 当且仅当 $a_{ik}=a_{kj}=1$，换句话说，就是从 v_i 到 v_k 和从 v_k 到 v_j 都有直接道路相通，所以 $a_{ij}^{(2)}$ 的值表示从 v_i 点出发经过某一个中间点 v_k，然后到 v_j 的路径数目，或形象地说 $a_{ij}^{(2)}$ 是从 v_i 出发两步到达 v_j 的路径数目。例如，$a_{42}^{(2)}=2$ 表示从 v_4 出发两步到达 v_2 的路径有两条，从图 1-1-3 中不难看出有 $v_4\to v_1\to v_2$ 和 $v_4\to v_3\to v_2$，而且仅有这两条。

明白了上述道理，要求从 v_1 点出发经 k 步所能到达的点，只需求矩阵 A^k 的第一行元素即可。A^k 的第一行行向量用 $A_1^{(k)}$ 表示。求 $A_1^{(k)}$ 只需 $A_1^{(k-1)}$，故若只计算 $A_1^{(k)}$，可大大减少计算量，$k=2,3,\cdots,n$。

同理可知 $a_{ij}^{(k)}$ 的值表示从 v_i 出发，k 步到达 v_j 的路径数目($a_{ij}^{(k)}=0$ 表示不存在这样的路径)若要追问这一路径是什么？它沿途经过哪几个点？只要回溯 $a_{ij}^{(k)}$ 这个数是怎么形成

的即可。

例如 $a_{17}^{(3)}=1$，我们来看看 $a_{17}^{(3)}$ 的形成过程：

图 1-1 4 中元素下标记录所经过的路径，如下标 15 即从 $v_1\to v_5$ 的道路，其它类似。

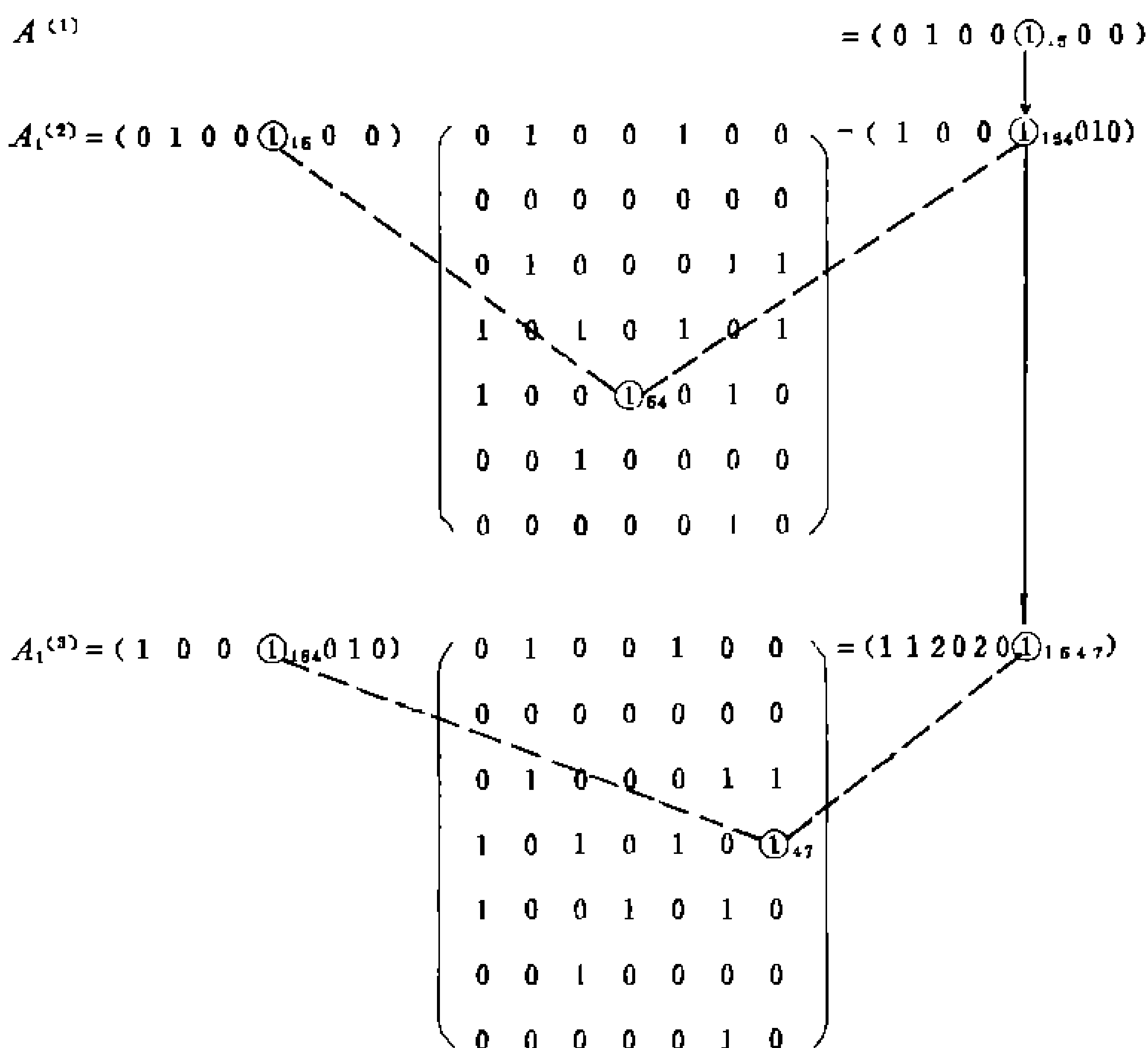


图 1

从图 1-1-4 可见，$A_1^{(3)}$ 中的 $a_{17}^{(3)}$ 由 $A_1^{(2)}$ 中的 $a_{14}^{(2)}$ 与 a_{47} 相乘而得的，即从 v_1 出发两步到达 v_4（即 $a_{14}^{(2)}=1$），再走一步可到 v_7。而 $a_{14}^{(2)}$ 是由 $A_1^{(1)}$ 中 $a_{15}^{(1)}$ 与 $a_{54}^{(1)}$ 相乘而得的，即从 v_1 出发第一步到 v_5，第二步才到 v_4。故由 $a_{15}^{(1)}\to a_{14}^{(2)}\to a_{17}^{(3)}$ 可知，这条路径是 $v_1\to v_5\to v_4\to v_7$。

假若要问从 v_7 到 v_1 是否有路相通？如上法所示，继续求 A^4，A^5，A^k，观察 $a_{71}^{(k)}$ 什么时候出现非零元素，即 k 为何值时，$a_{71}^{(k)}\neq 0$，而且 k 的值不能超过 6，即 $k<7$。七个顶点，若从其中一点出发（设为 v_1），走了六步还到不了 v_7，则再走下去也到不了。这个道理留给大家思考。

对于要证明 v_7 没有道路走到 v_1，还可以采用如此证法。若把点的次序重新排列为：$v_7,v_6,v_3,v_2,v_5,v_4,v_1$，可得

$$
A = \begin{matrix} v_7 \\ v_6 \\ v_3 \\ v_2 \\ v_5 \\ v_4 \\ v_1 \end{matrix}
\left(\begin{array}{cccc:ccc}
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 1 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0
\end{array}\right)
= \begin{pmatrix} A_{11} & O \\ A_{21} & A_{22} \end{pmatrix},
$$

$$
\qquad\quad v_7 \quad v_6 \quad v_3 \quad v_2 \quad v_5 \quad v_4 \quad v_1
$$

其中

$$
A_{11} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A_{21} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},
$$

$$
A_{22} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \qquad A_{12} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}。
$$

$$
A^2 = \begin{pmatrix} A_{11} & O \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} A_{11} & O \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} A_{11}^2 & O \\ A_{21}A_{11} + A_{22}A_{21} & A_{22}^2 \end{pmatrix} = \begin{pmatrix} A_{11}^{(2)} & O \\ A_{21}^{(2)} & A_{22}^{(2)} \end{pmatrix}。
$$

而且不难证明，对于任何整数 k 恒有

$$
A^k = \begin{pmatrix} A_{11}^{(k)} & O \\ A_{21}^{(k)} & A_{22}^{(k)} \end{pmatrix}, k = 1, 2, \cdots。
$$

这就证明 $a_{71}^{(k)} = 0$，即从 v_7 出发不论走多少步都到不了 v_1。实际上它们之间由于 $A_{12}^{(k)} = O$，说明 v_7, v_6, v_3, v_2 点与 v_5, v_4, v_1 点不通，即没有这样的道路。

这个例子只是讨论从一点到另一点有没有道路相通？有几条道路相通？若存在几条道路相通，再进一步便是寻找其中一点到另一点的最短路径问题，这是一个很有实际意义的运筹学问题。后面将讨论它。

三、路径问题应用举例

若我方两名军事人员和敌方两名军事人员同到某一现场视察，途中要经过一条河。现只有一只小船，每次最多只能乘两个人。为了安全起见，当敌我双方人员同时在场时应避免出现我方人员少于敌方人员的情况，问渡河的方案应如何？

为方便起见，设用 (m, n, l) 表示河的左岸有我方人员 m 人，敌方人员 n 人的状态；(m, n, r) 表示在河的右岸我方人员 m 人，敌方人员 n 人的状态。显然可见，若左岸为 (m, n, l) 的状态，则右岸的状态为 $(2-m, 2-n, r)$。

现将全部可能的状态列举如下：

$v_1=(2,2,l)$, $v_7=(2,2,r)$,

$v_2=(2,1,l)$, $v_8=(2,1,r)$,

$v_3=(1,1,l)$, $v_9=(1,1,r)$,

$v_4=(2,0,l)$, $v_{10}=(2,0,r)$,

$v_5=(0,2,l)$, $v_{11}=(0,2,r)$,

$v_6=(0,1,l)$, $v_{12}=(0,1,r)$。

请注意，这里不出现$(1,0,l)$，$(1,0,r)$状态。因表面上看我方人员多于敌方的情况。然而它的对岸则是$(1,2,r)$，$(1,2,l)$，这是不允许的。

渡船的全过程可以看作是状态的转移，则状态之间的关系，可用图 1-1-5 表示。其中点表示状态，状态间的转移用连线表示，比如从$v_1(2,2,l)$可以迁移到$v_{12}(0,1,r)$，只要从左岸运一敌方人员到右岸即可。(v_1,v_{12})联线上的(0,1)符号说明这关系。显然状态之间的关系是可逆的、对称的。也就是说v_1 可以转化到v_{12}，v_{12}也可以转化为v_1。相互转化的两状态之间用一无向边相连。如连接v_1 和 v_{12}等等。

原问题归结为从图 1-1-5 找一条从v_1 到v_7 的路径。且每条路径表达一种渡船的方案。

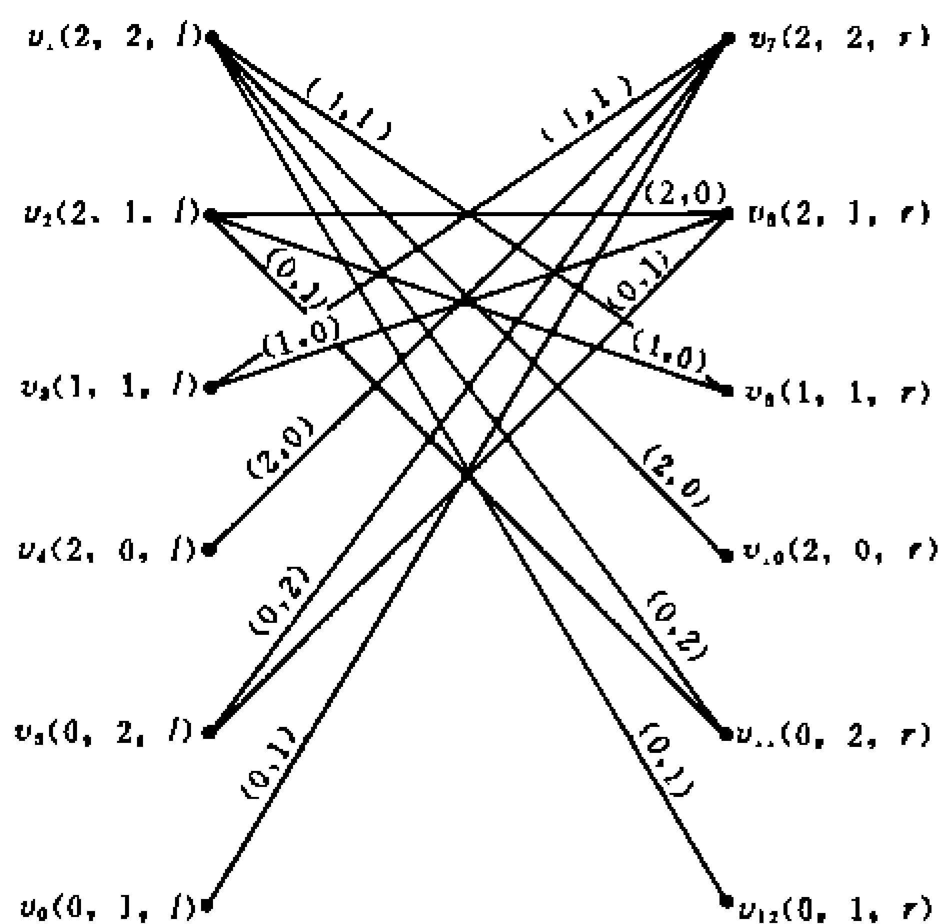
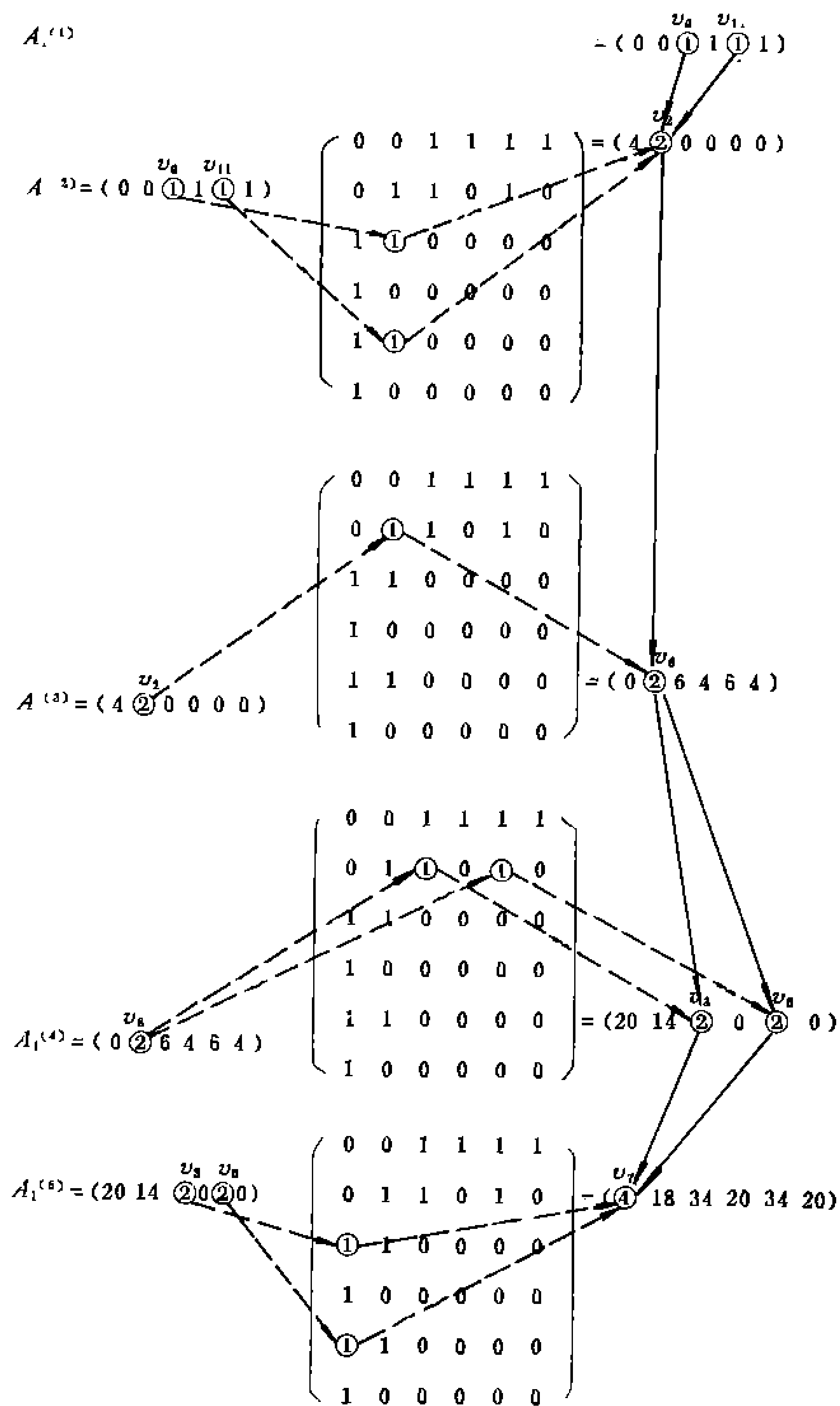


图 1-1-5

如前面例子一样可得形式如下的对称矩阵：

$$A=\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}=\begin{pmatrix} O & A_1 \\ \bar{A}_1 & O \end{pmatrix}_{12\times 12}$$

由于两岸状态的对称性，所以矩阵 A 也是对称矩阵。点的排列次序是按下标的自然次序，即$v_1,v_2,v_3,\cdots,v_{12}$，其中$A_{11}=A_{22}=O$，(转第 10 页)



A₁⁽¹⁾
v₆ v₁₁
= (0 0 ① 1 ① 1)
v₆ v₁₁
A₁⁽²⁾ = (0 0 ① 1 ① 1)
0 0 1 1 1 1
0 1 1 0 1 0
1 ① 0 0 0 0
1 0 0 0 0 0
1 ① 0 0 0 0
1 0 0 0 0 0
v₂
= (4 ② 0 0 0 0)
0 0 1 1 1 1
0 ① 1 0 1 0
1 1 0 0 0 0
1 0 0 0 0 0
1 1 0 0 0 0
1 0 0 0 0 0
v₂
A₁⁽³⁾ = (4 ② 0 0 0 0)
v₆
= (0 ② 6 4 6 4)
0 0 1 1 1 1
0 1 ① 0 ① 0
1 1 0 0 0 0
1 0 0 0 0 0
1 1 0 0 0 0
1 0 0 0 0 0
v₆
A₁⁽⁴⁾ = (0 ② 6 4 6 4)
v₃ v₅
= (20 14 ② 0 ② 0)
0 0 1 1 1 1
0 1 1 0 1 0
① 1 0 0 0 0
1 0 0 0 0 0
① 1 0 0 0 0
1 0 0 0 0 0
v₃ v₅
A₁⁽⁵⁾ = (20 14 ② 0 ② 0)
v₇
= (④ 18 34 20 34 20)

图 1-16

$$A_2 = A_{21} = A = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A^2 = \begin{bmatrix} O & \bar{A}_1 \\ \bar{A}_1 & O \end{bmatrix}\begin{bmatrix} O & \bar{A}_1 \\ A_1 & O \end{bmatrix} = \begin{bmatrix} A_1^2 & O \\ O & \bar{A}_1^2 \end{bmatrix},$$

$$\bar{A}^3 = \begin{bmatrix} \bar{A}_1^2 & O \\ O & \bar{A}_1^2 \end{bmatrix}\begin{bmatrix} O & \bar{A}_1 \\ \bar{A}_1 & O \end{bmatrix} = \begin{bmatrix} O & A^3 \\ \bar{A}_1^3 & O \end{bmatrix},\cdots。$$

请读者说明上述规律的实际意义。

现在问题归结为求 $A, A^3, \bar{A}^5, \cdots$，设 $\bar{A}^k = (a_{ij}^{(k)})$，并注意什么时候出现 $a_{ij}^{(k)} \neq 0$。但 k 为奇数时使 $a_{ij}^{(k)} \neq 0$ 才是所求的。

令 $A_1^{(k)}$ 表由矩阵 $\bar{A}^k$ 中第一行元素组成的行向量。运算过程见第 9 页图 1-1-6。从运算结果分析，可得从 v_1 到 v_7 的四条路径如下：

1. $v_1 \to v_9 \to v_2 \to v_8 \to v_3 \to v_7$，
2. $v_1 \to v_9 \to v_2 \to v_8 \to v_5 \to v_7$，
3. $v_1 \to v_{11} \to v_2 \to v_8 \to v_3 \to v_7$，
4. $v_1 \to v_{11} \to v_2 \to v_8 \to v_5 \to v_7$。

或用图 1-1-7 表示。

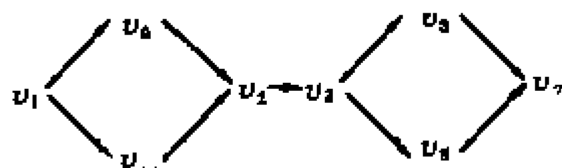


图 1-1-7

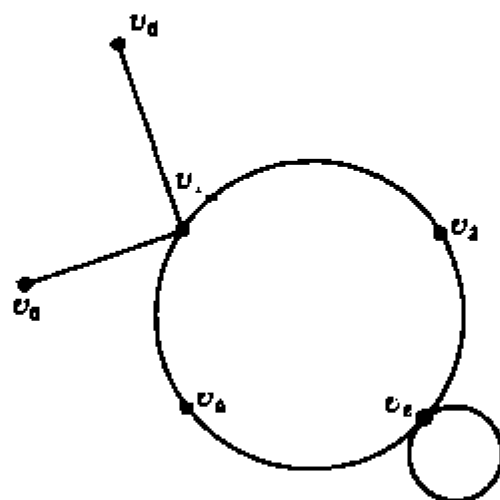


图 1-1-8

上面讨论的方法还可进一步简化。比如用一个顶点表示(h,k)状态，即我方 h 人，敌方 k 人的一种状态，不区分它究竟是左岸还是右岸，这是可行的。因为河的左岸和右岸是交替出现的，状态迁移图可用图(1-1-8)来表示。

其中 $v_1=(2,2)$，$v_2=(2,1)$，$v_3=(1,1)$，$v_4=(0,2)$，$v_5=(2,0)$，$v_6=(0,1)$。

特别要指出的是，v_2 点有一从 v_2 点到 v_2 的自环，它表示从(2,1)状态回到(2,1)状态，但已从河的一岸的(2,1)状态迁移到河的另一岸的(2,1)状态。这样的圆环我们称为自环，或简称为环。图 1-1-8 是无向图，实际上每条边都是双向的。

同样的道理，从 $v_1(2,2)$点出发走一步到 $v_6(0,1)$，便是从河岸(2,2)状态转移到对岸的(0,1)状态。于是问题导致求从 v_1 出发，经过奇数步返回 v_1 的路径。

另外，由图 1-1-8 可见，从 v_1 到 v_4，v_4 只能有一种可能返回 v_1，故 v_5 可省略不考虑。v_6 也是一样。所以问题就大大地简化了，导致求图 1-1-9 中从 v_1 出发，经奇数步重返 v_1 的

路径。

只要通过观察图 1-1-9 就可以很快给出问题的解答。它应该是：

1. $v_1 \to v_4 \to v_2 \to v_2 \to v_4 \to v$ ，

2. $v_1 \to v_4 \to v_2 \to v_2 \to v_3 \to v$ ，

3. $v \to v_3 \to v_2 \to v_2 \to v_4 \to v$ ，

4. $v \to v_3 \to v_2 \to v_2 \to v_3 \to v$ 。

或如图 1-1 10 所示。

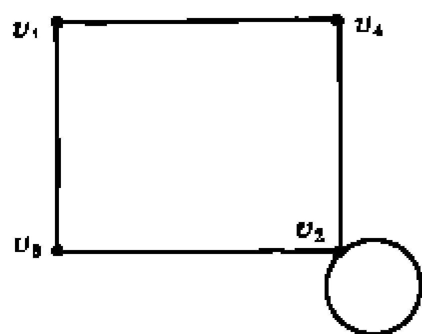


图 .1-9

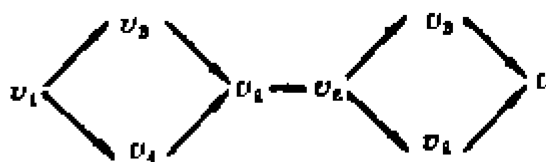


图 1-1 1)

通过计算也可得出相同结果。为此令：

$$A = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix}\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}_{\begin{matrix} v & v_2 & v_3 & v_4 \end{matrix}},$$

计算 A^k，$k=3,5,\cdots$，直到出现 $a^{(k)}$ 不为零为止。按照前面的方法计算如图 1-1 11 所示。

四、Hamilton 回路问题

Hamilton 回路是以 1856 年 Hamilton 首先提出的所谓环球航行问题而得名。如图 1-1 13 所示，20 个顶点分别表示世界 20 个名城，两个顶点间的连线表示这两个城市间的航线。要求旅行者从某一城市出发，遍历各城市一次且仅一次，最后返回出发点。

Hamilton 回路问题，不同于 Euler 回路问题，它是求对顶点的遍历，在运筹学中有着实际意义。特别是求 Hamilton 回路中总距离最短的问题，是有名的旅行商人问题，后面将讨论它。

图 1 1 12 也可以看作是十二面体(见图 1 1 13)，每一个面都是五边形。沿着十二面体的棱，到达每一个顶点时，都面临着两种道路的选择：一是向左，另一是向右。向左的道路用 L 表示，向右的道路用 R 表示。LR 表示第一步向左转，第二步向右转，其它同理类推。

这一个 Hamilton 问题的解决颇有趣味，现介绍如下。由于有：

(a) $(LL)R=L(LR)$；

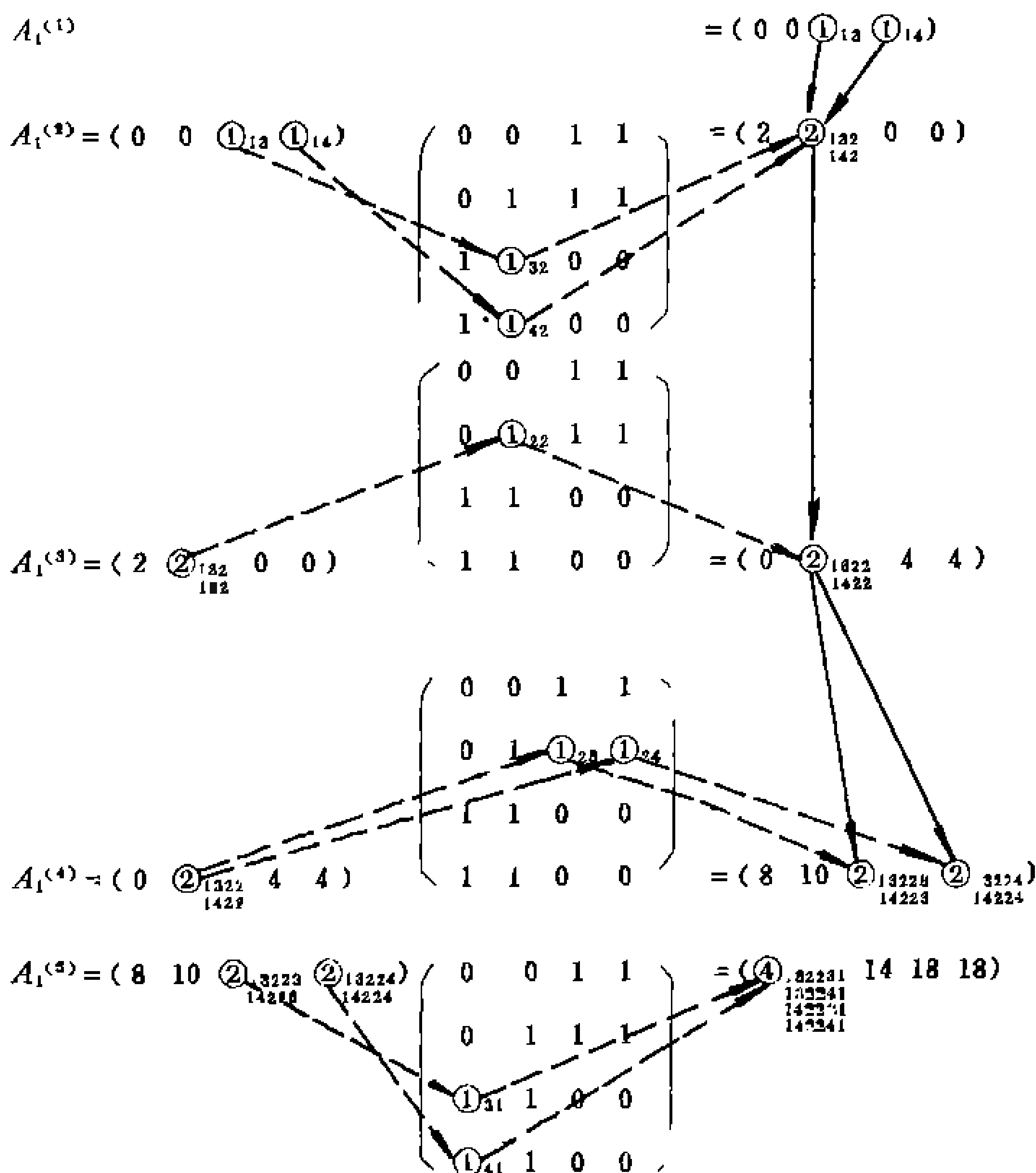


图 1-1-11

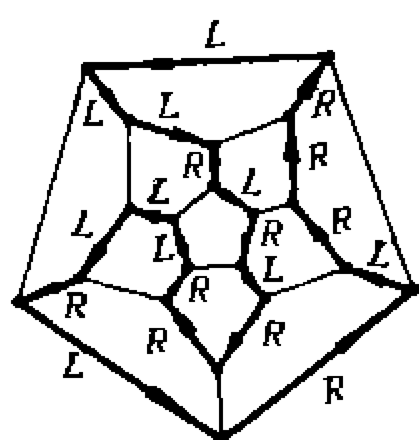


图 1-1-12

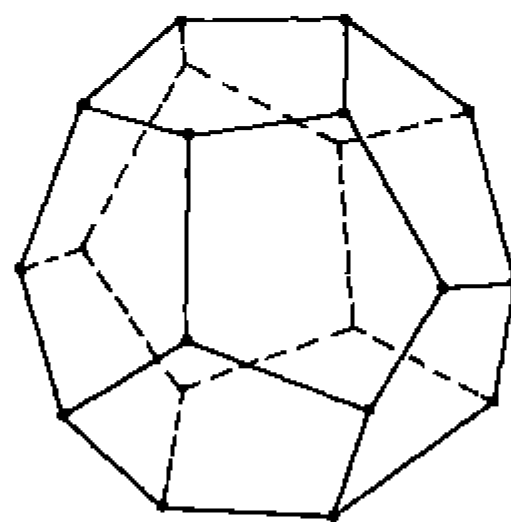


图 1-1-13

(b) $R^5=L^5=1$;

(c) $RL^2R=LRL$, $LR^2L=RLR$;

(d) $RL^3R=L^2$, $LR^3L=R^2$。

$L^5=1$ 的意义是连续向右转 5 次回到原来的出发点。不难直接验证上面等式是正确的。

依据上面的公式可得下面的推导：

$$1=R^5=R^2R^3 \xlongequal{\text{由}(d)} (LR^3L)\cdot R^3=(LR^3)(LR^3)=(LR^3)^2$$
$$=(LR^2R)^2 \xlongequal{\text{由}(d)} [L(LR^3L)R]^2=[L^2R^3LR]^2$$
$$\xlongequal{\text{由}(d)} [L^2(LR^3L)RLR]^2=[L^3R^3LRLR]^2$$
$$=[LLLRRRLRLR]^2$$
$$=LLLRRRLRLRLLLRRRLRLR。$$

该式右端共 20 项，左端归约为 1。说明沿右端的规律走了 20 步后回到原地，特别是右端的若干部分项没有归约为 1 的可能性，故从任一点出发沿右端的规律而遍历每一点一次返回原地。

五、计算机程序的流程图

一般编写程序以前都须先画出程序的流程图，例如用简单的办法"判定大于 1 的整数 i 是否是素数"的流程图如图 1-1-14 所示。

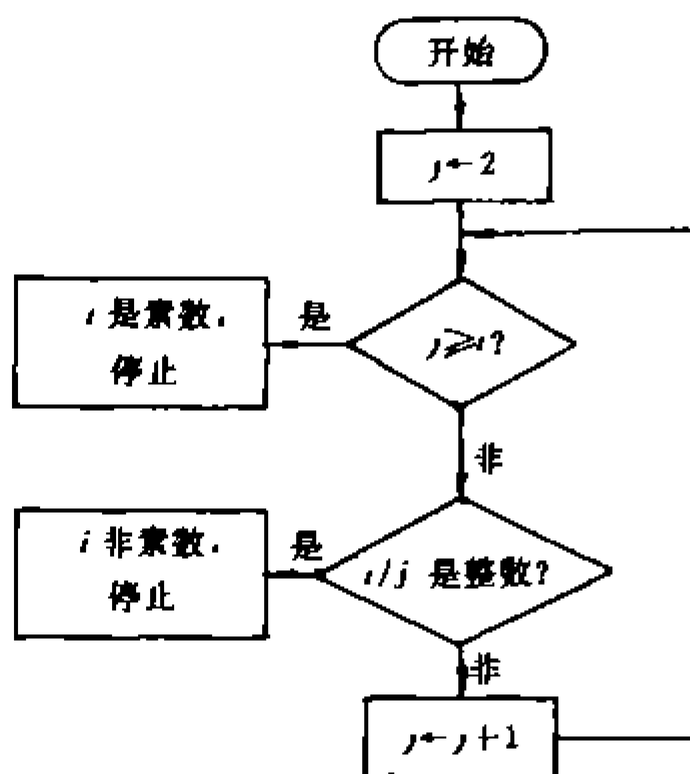


图 1-1-14

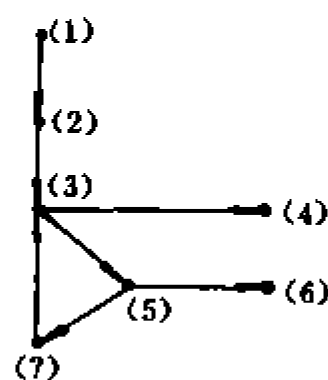


图 1-1-15

如果略去方框中的具体内容，将每个方框抽象成一个点，则流程图可抽象成图 1-1-15 的有向图。实际上任何一个程序流程图，都可用一个有向图来刻划。将程序转化为程序流程图以后，问题变成讨论流程图的性质，它形成"数据流图"一个新的分支，它的讨论超出

了本书的范围。但利用图可直观地判断程序的逻辑是否正确，是否存在死循环，对操作系统而言，则可用来判断是否存在死锁现象。总之，图论在计算机科学中扮演了非常重要的角色。

六、Ramsey 问题

问题是这样的：任意 6 个人在一起，6 人中要不是有 3 个人彼此互相认识，必然有 3 个人互相不认识；即两种情况中至少存在一种。

用图的办法很容易给 Ramsey 问题以说明。设这 6 个人分别用 v_1,v_2,v_3,v_4,v_5,v_6 6 个点表示。互相认识的两个人，其对应顶点用实线相连，互不相识的两个人，其对应顶点用虚线相连。两个人要么认识，要么互不相识，两者必居其一。所以，这 6 个顶点中任何一点，与其它任何一点必有一连线（实线或虚线）。这样的图叫完全图。问题变为由此所得的 6 个顶点的完全图中，至少存在一个实线三角形或虚线三角形。

为说明结论，任取一点 v_1，它与其它 5 点的联线中，至少有 3 条同为实线或同为虚线。不妨假设同为实线，而这 3 条线的另一端点分别为 v_i,v_j,v_k。这 3 个顶点形成的三角形，若是虚线三角形，则问题得到解决。如若不然，至少有一条边设为 v_iv_j 为实线，则 $v_1v_iv_j$ 便是一实线三角形（图 1-1-16）。

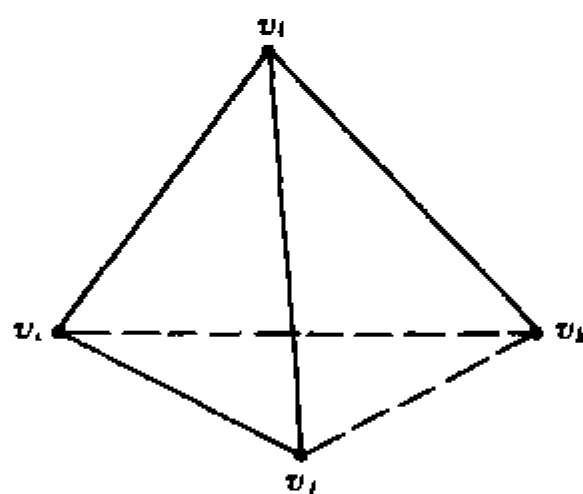


图 1-1-16

所以 Ramsey 问题得到证明。

这一节给出了一些利用图使问题变得直观简单的实例，给大家一个图论研究的直观印象，下面我们介绍图论的一些基本概念。

§2 图的概念

一般几何上将图定义成空间一些点（顶点）和连接这些点的线（边）的集合。

图论中将图定义为一个偶对 $G=(V,E)$，其中 V 表示顶点的集合，E 表示边的集合。这样如图 1-2-1 可表示为

$$V=\{v_1,v_2,v_3,v_4\},\ E=\{e_1,e_2,e_3,e_4,e_5,e_6\}。$$

我们也可以用边的两个顶点来表示边。如果边 e 的两个端点是 u 和 v，那么 e 可写成 $e=\langle u,v\rangle$，这儿 $\langle u,v\rangle$ 表示 u 和 v 的无序对。即 $\langle u,v\rangle$ 和 $\langle v,u\rangle$ 都表达了以 u,v 为端点的无向边。若不引起理解上混乱，通常也用 (u,v) 来表示 $\langle u,v\rangle$，这样图 1-2-1 可写成：

$$G=(V,E),\quad V=\{v_1,v_2,v_3,v_4\},$$

$$E=\{\langle v_1,v_2\rangle,\langle v_1,v_3\rangle,\langle v_1,v_4\rangle,\langle v_2,v_3\rangle,\langle v_2,v_4\rangle,\langle v_3,v_4\rangle\}。$$

一般图 $G=(V,E)$ 的顶点数用 $n(=|V|)$ 表示，边的数目用 $m(=|E|)$ 表示。若 $|V|$ 和 $|E|$ 都是有限的，则称图 G 是有限图，否则称为无限图。本书只讨论有限图的情况。

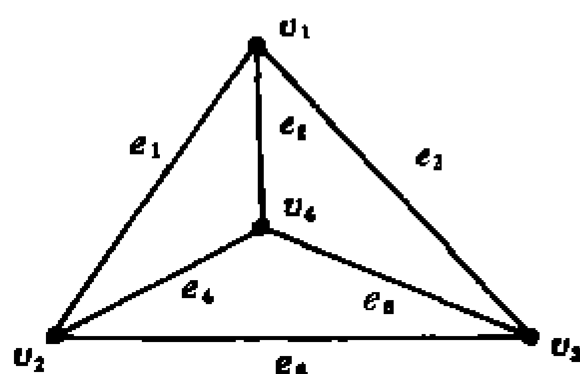


图 1-2-1

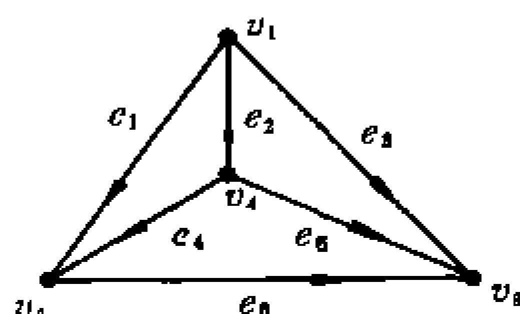


图 1-2-2

上面讨论的图G的边的两个顶点是无序的，一般称其为无向图，在实际应用中，将图和每条边分配一个方向是很自然的。当给图G的每一条边规定一个方向，则称其为有向图。对有向图$G=(V,E)$，有向边e用与其关联的顶点u,v的有序对来表示，即$e=(u,v)$，表示u为边e的起点，v为边e的终点。那么图1-2-2所示的有向图可表示如下：

$$G=(V,E),\quad V=\{v_1,v_2,v_3,v_4\},$$

$$E=\{(v_1,v_2),(v_1,v_3),(v_1,v_4),(v_4,v_2),(v_4,v_3),(v_2,v_1)\}。$$

如果顶点v是边e的一个端点，则称边e和顶点v相关联(incident)；对于顶点u和v，若$(u,v)\in E$，则称u和v是邻接的(adjacent)。在图1-2-1中，边e_3,e_4,e_5都与顶点v_4相关联，v_4分别与v_1,v_2,v_3相邻接。若两条边有共同的顶点，则称这两条边是邻接的。图1-2-1中，边e_1,e_2,e_3两两相邻接。

对图$G=(V,E)$和$G'=(V',E')$来说，若有$V'\subseteq V$和$E'\subseteq E$，则称图G'是G的一个子图；若$V'\subset V$或$E'\subset E$，则称图G'是G的一个真子图。

对无向图$G=(V,E)$，定义：

$$Inc(v_i)\triangleq\{e_k|e_k=(v_i,v_j)\in E\},$$

$$Adj(v_i)\triangleq\{v_j|e_k=(v_i,v_j)\in E, v_i\neq v_j\},$$

即$Inc(v_i)$表示以v_i为一顶点的边(即与v_i关联的边)的集合，$Adj(v_i)$表示与顶点v_i相邻接的顶点的集合。

对有向图$G=(V,E)$，定义：

$$Inc^+(v_i)\triangleq\{e_k|e_k=(v_i,v_j)\in E\},$$

$$Inc^-(v_i)\triangleq\{e_k|e_k=(v_j,v_i)\in E\}。$$

即$Inc^+(v_i)$表示以v_i为始点的有向边的集合，

$Inc^-(v_i)$表示以v_i为终点的有向边的集合。

同样我们也可定义：

$$Adj^+(v_i)\triangleq\{v_j|e_k=(v_i,v_j)\in E\};$$

$$Adj^-(v_i)\triangleq\{v_j|e_k=(v_j,v_i)\in E\}。$$

例：如图1-2-3有

$$Inc(v_1)=\{e_1,e_2,e_3\},\quad Adj(v_1)=\{v_2,v_3\};$$

$$Inc(v_2)=\{e_1,e_4,e_5\},\quad Adj(v_2)=\{v_1,v_3\};$$

$$Inc(v_3)=\{e_2,e_4,e_5\},\quad Adj(v_3)=\{v_1,v_2\}。$$

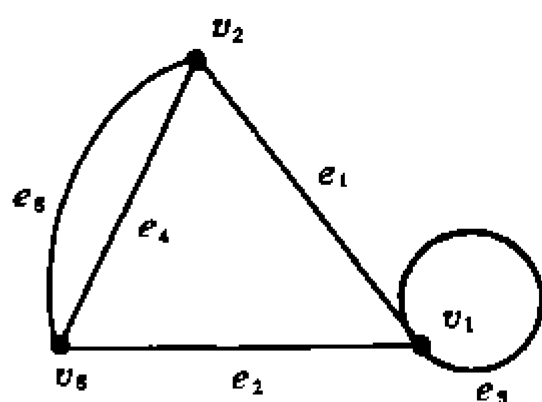


图 1-23

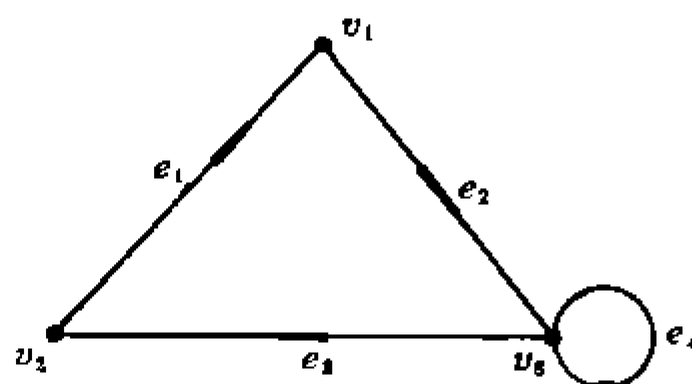


图 1-24

例：如图 1-24 有

$$Inc^+(v_1) = \{e_2\},$$
$$Inc^-(v_1) = \{e_1\},$$
$$Inc^+(v_3) = \{e_3, e_4\},$$
$$Inc^-(v_3) = \{e_2, e_4\}。$$

定义 设$G=(V,E)$是无向图，若顶点v_k是G的一个顶点，且不存在自环。则$d(v_k) \triangleq |Inc(v_k)|$。称为顶点$v_k$的度。这里$|Inc(v_k)|$是集合$Inc(v_k)$中元素个数。即$d(v_k)$表示$G$图中以$v_k$为端点的边数，或说是与$v_k$点发生关联的边数。

定义 对于有向图$G=(V,E)$，v_k是G的一顶点，则

$$d^+(v_k) \triangleq |Inc^+(v_k)|,$$
$$d^-(v_k) \triangleq |Inc^-(v_k)|。$$

分别称为顶点v_k的出度和入度。

$$d(v_k) \triangleq d^+(v_k) + d^-(v_k)$$

称为顶点v_k的度。

显然若v_k是有向图的顶点，而且有自环时，则此自环既从v_k出发，又进入v_k，故计算了两次。为统一起见，对无向图G中有自环的顶点v_k，此自环在$d(v_k)$中也应计算两次。

如在图 1-1-13 中，$d(v_1)=4$，$d(v_2)=d(v_3)=3$。在图 1-1-14 中$d^+(v_1)=1$，$d^-(v_1)=1$，$d(v_1)=2$。

由上面的讨论，很容易得出如下结论：

结论 1 对于图$G=(V,E)$恒有

$$\sum_{v_i \in V} d(v_i) = 2|E|。$$

证明留给读者自己来完成。

结论 2 对于任意图$G=(V,E)$，度为奇数的点数必为偶数。

证明 从上面结论 1 可知，各顶点度的和$\sum_{v_i \in V} d(v_i)$为偶数。设度为偶数的顶点集合为V_e，度为奇数的顶点集合为V_o，则：

$$V = V_e \cup V_o,$$
$$\sum_{v_i \in V_e} d(v_i) + \sum_{v_i \in V_o} d(v_i) = 2|E|。$$

显然和 $\sum_{v_i \in V_e} d(v_i)$ 为偶数。因为偶数的和为偶数。若 $|V_o|$ 为奇数，则 $\sum_{v_j \in V_o} d(v_j)$ 必为奇数。因为奇数个奇数的和必为奇数，奇数与偶数的和为奇数，则与结论 $\sum_{v_i \in V_e} d(v_i) + \sum_{v_j \in V_o} d(v_j) = 2|E|$ 相矛盾。故 $|V_o|$ 必为偶数。

§3 道路与回路

一、道路与回路

定义 图 $G=(V,E)$的一个顶点和边的交替序列 $\mu=v_0e_1v_1\cdots v_{k-1}e_kv_k$，且边 e_i 的端点为 v_{i-1} 和 v_i，$i=1,2,\cdots,k$，则称 μ 为一条道路(Path)，v_0 和 v_k 分别称为 μ 的起点和终点。特别如 μ 中所有的边均不相同，则称其为简单道路。以 v_0 为起点，v_k 为终点的道路称为 v_0-v_k 道路。

如果道路 μ 中 $v_0=v_k$，称 μ 为回路，回路中没有重复边时称为简单回路。

对于有向图也可类似定义道路、回路的概念，留给读者思考。

例：

在图 1-3-1 中，$S=(v_2e_1v_1e_5v_3e_6v_4)$ 是一道路，$C=v_1e_2v_2e_3v_3e_6v_4e_4v_1$ 是一回路。

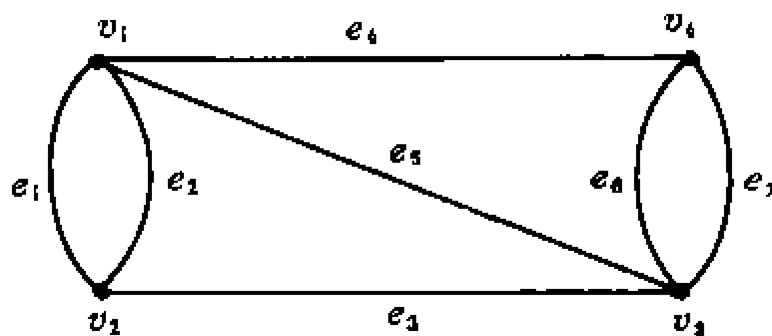


图 1-3-1

定义 对图 $G=(V,E)$来说，若 G 的两顶点 u，v 之间存在一条道路；则称 u 和 v 是连通的，若图 G 的任意两顶点连通，则称图 G 是连通的；否则是非连通的。非连通图可分解为若干连通子图。

对于有向图，若边去掉方向后是连通的，则称其为连通的有向图。若对于有向图的任意两顶点 u 和 v 之间存在 u 到 v 道路和 v 到 u 道路时，称其为强连通的。

定义 没有重复边和自环的图叫做简单图。

二、欧拉(Euler)回路

定义 对于连通的无向图 G，若存在一简单回路，它通过 G 的所有边，则这回路称为 G 的 Euler 回路。

定理 若连通无向图 G 的所有顶点的度都是偶数，则存在一条图 G 的 Euler 回路。

证明 不失一般性，设从 v_0 出发，边数最多的一条回路为

$$C=\{e_1=(v_0,v_1),e_2=(v_1,v_2)\cdots,e_m=(v_m,v_0)\}。$$

我们将证明回路 C 是 G 的 Euler 回路。

证明 方法是构造性的，即实际上提供一种求 Euler 回路的一种算法。假定 C 不是 Euler 回路，则 C 至少有一顶点，该顶点的度大于回路 C 过该点的边数。如若不然，G 中不属于 C 的任一边，其两端点也不属于 C，这与连通图的假设矛盾。这就证明了如果最大回路 C 不是 Euler 回路，则 C 中至少有一点（设为 v_k），使得回路 C 过 v_k 的边数小于 $d(v_k)$。令 $v_k = v_1'$，从 v_1' 出发有一不属于 C 的边 $\bar{e} = (v_1', v_2')$。由于 $d(v_2')$ 是偶数，若 $v_2' \neq v_1'$，则有不属于 C 的边 $\bar{e} = (v_2', v_3')$，依次类推，不属于 C 的边 $e_3 = (v_3', v_4')$，…，$e = (v_{l-1}', v_l')$，使得边 e_l 的终点 $v_l' = v_1'$。故 $C_1 = (e, e, \cdots, e_l)$ 是一条新回路。这样 C 与 C_1 构成回路 $C' = (e_1, e_2, \cdots, e_k, e_1', e', \cdots, e_l', e_{k+1}, \cdots, e_m)$，这与 C 是最大回路的假设相矛盾。故 C 是 Euler 回路。

推论 如果连通图 G 只有两个度为奇数的顶点，则存在一条以这两个顶点为两端点，包含 G 的所有边的简单道路。这一道路称之为 Euler 道路。

证明 设 v_1, v_2 是两个度为奇数的点，对图 G 加上一条虚边 $e_0 = (v_1, v_2)$ 得一新的图 G'。由于 G' 的所有顶点的度都是偶数，根据定理，G' 存在 Euler 回路 $C' = (e_0, \cdots)$，e_0 是 C' 的最开始的边，C' 中除去 e_0 便得从 v_2 到 v_1 的道路 P，P 满足推论所要求的条件。

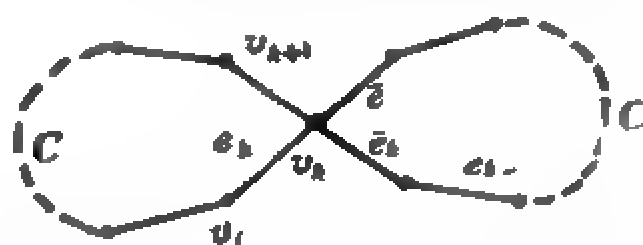


图 1-3

模数转换问题：

如图 1-3-3 所示，旋转鼓的面分成 16 部分，设鼓的位置用 4 位二进制数 $abcd$ 表示，用导与不导材料表示 1 和 0 两种状态，4 个相连的扇形给出一位二进制数。试问这 16 个扇形应如何排列便能使得相邻 4 个扇形（共 16 组）组成 0000 到 1111 的 16 个数。

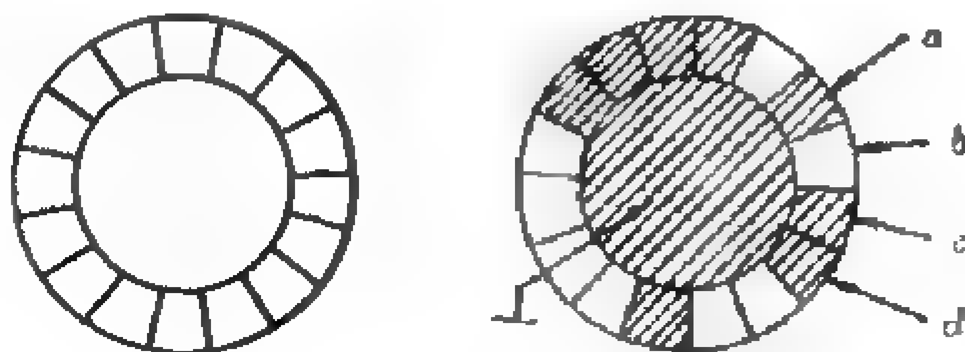


图 1-3-3

如图 1-3-4 所示，8 个顶点分别表示 000 到 111 的 8 个数。从顶点 $a_1a_2a_3$ 到 $a_2a_3a_4$ 引一线段，表示 $a_1a_2a_3a_4$ 这 4 位二进制数。最后问题导致求上 DeBruijn 图（图 1

3-4)的一条 Euler 回路。图 1-3-4 或称为 $n=3$ 的 DeBruijn 图。图 1-3-4 的 8 个顶点的度都为偶数，入度和出度均为 2。故存在 Euler 回路。例如，$\{e_0,e_1,e_2,e_5,e_{10},e_4,e_9,e_3,e_6,e_{13},e_{11},e_7,e_{15},e_{14},e_{12},e_8\}$ 便是一条 Euler 回路，对应于这一回路的有序列：

0000101001101111000。

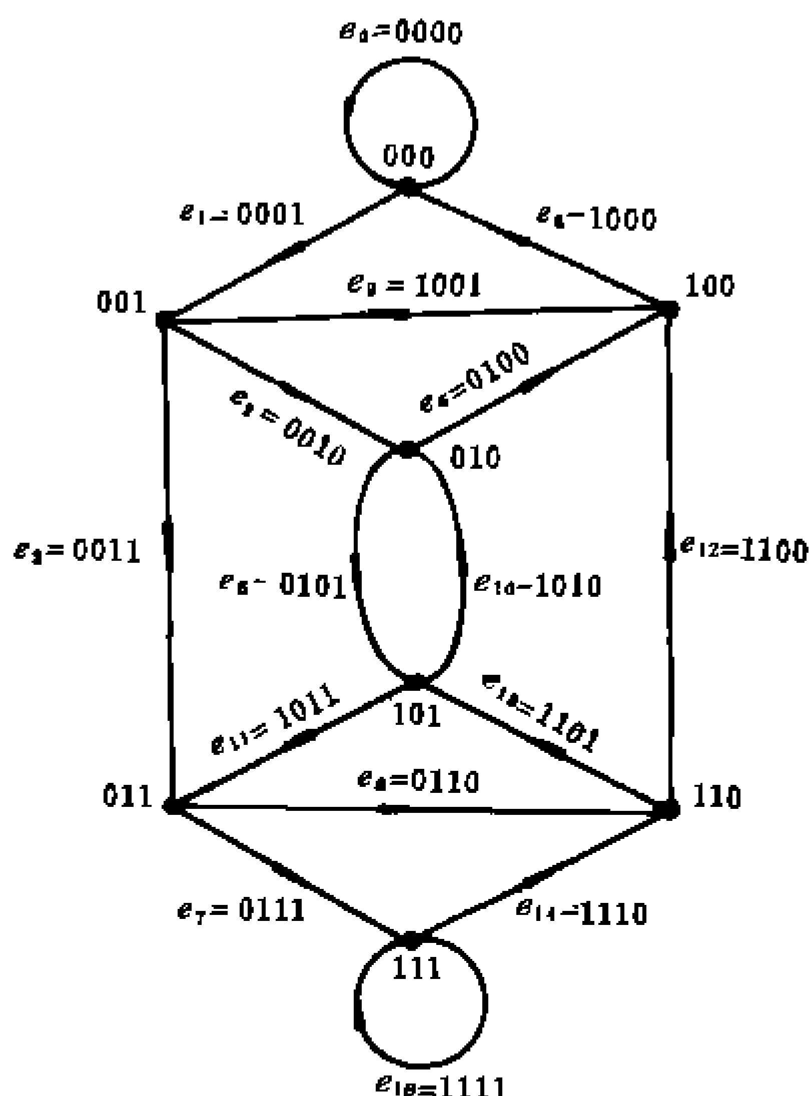


图 1-3-4

这 19 个比特的序列任一相邻 4 位数表达了从 0000 到 1111 的 16 个数。例如 $e_0 \leftrightarrow 0000$，$e_1 \leftrightarrow 0001$，$e_2 \leftrightarrow 0010$，$e_5 \leftrightarrow 1010$，$e_{10} \leftrightarrow 0100$，…，$e_8 \leftrightarrow 1000$。

三、哈密尔顿(Hamilton)回路

哈密尔顿(Hamilton)回路的概念已见于第一节。若图 G 存在一条道路 P，它通过每一顶点各一次，则称 P 为图 G 的 Hamilton 道路。

定理 设简单图 G 的顶点数为 $n(n>3)$，若 G 中任意一对顶点 v_i,v_j，恒有 $d(v_i)+d(v_j)\geqslant n-1$，则图 G 至少有一条 Hamilton 道路。

又若对任意一对 v_i,v_j，$d(v_i)+d(v_j)\geqslant n$，则存在一条 Hamilton 回路。

证明 (a)先证图 G 一定是连通的。如若不然，G 至少有两个互不连通的部分。设其中之一有 n_1 个顶点，另一部分有 n_2 个顶点。分别从中各选一顶点 v_1,v_2，由于图 G 是简单图，故有

$$d(v_1)\leqslant n_1-1,\quad d(v_2)\leqslant n_2-1,$$
$$d(v_1)+d(v_2)\leqslant n_1+n_2-2<n-1。$$

这和假定 $d(v_i)+d(v_j)\geqslant n-1$ 发生矛盾。故证图 G 是连通的。

(b)再证明存在 Hamilton 道路。证明的方法实际上给出一种构造 Hamilton 道路的步骤。不失一般性，不妨设如果图 G 有一条从 v_1 到 v_l 的道路$(v_1,v_2,\cdots,v_l)$，设这条道路两端点无法向外延伸，即这条道路的两端点 v_1 和 v_l 只和这条道路中的点邻接，不和此外的点相邻接，那么 $v_1,v_2,\cdots,v_l$ 点之间一定存在一条回路。假定 v_1 点邻接于点 $v_{j_1},v_{j_2},\cdots,v_{j_k}$，这里 $v_{j_1},v_{j_2},\cdots,v_{j_k}$都是道路$(v_1,v_2,\cdots,v_l)$中的点，而且 $l<n$。

如果 v_1 邻接于 v_l 点，则显然存在回路

$$(v_1,v_2,\cdots,v_l,v_1)。$$

如果 v_1 不和 v_l 相邻，则必然存在一 v_p 点$(2\leqslant p\leqslant l)$和 v_1 相邻，而 v_{p-1}和 v_l 相邻接，如图 1-3-5 图所示。因为否则 v_l 最多只和 $l-k-1$ 个顶点相邻接，即排除

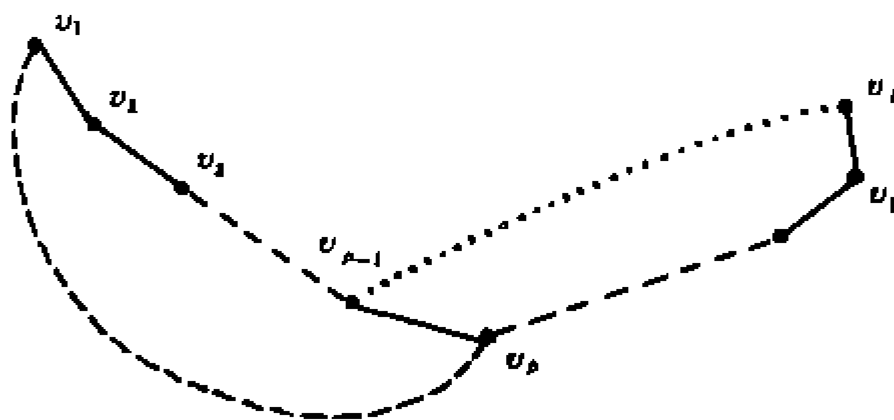


图 1-3-5

$v_{j_1-1},v_{j_2-1},\cdots,v_{j_k-1}$和 v_l 自身，这样有

$$d(v_1)+d(v_l)\leqslant k+(l-k-1)=l-1<n-1,$$

和假设矛盾。因而存在过 $v_1,v_2,\cdots,v_l$ 的回路(图 1-3-6)

$$(v_1,v_p,v_{p+1},\cdots,v_l,v_{p-1},v_{p-2},\cdots,v_2,v_1)。$$

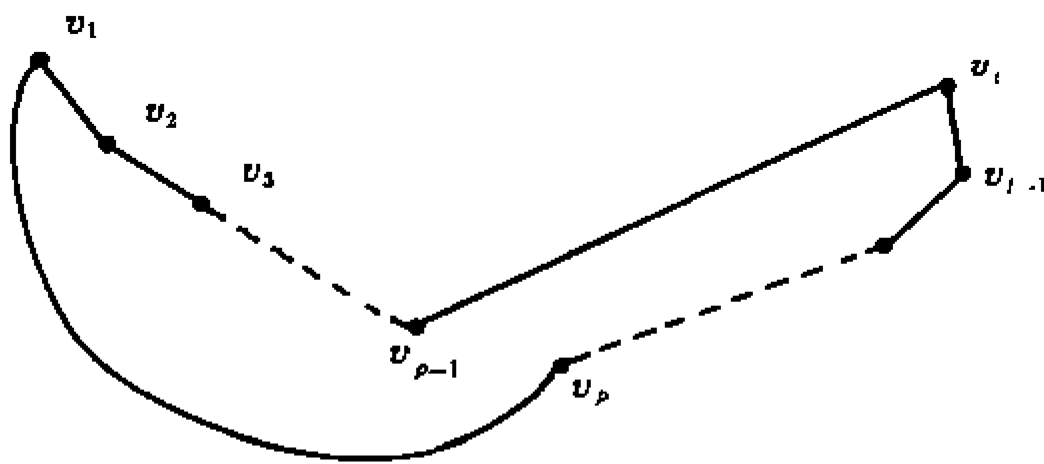


图 1-3-6

总之，从 v_1 到 v_l 有一条道路 $v_1,v_2,\cdots,v_l$，则必存在一条过这 l 点的回路。

若 $l=n$，实际上已存在一条 Hamilton 道路。定理得到证明。

若 $l<n$，由于已证 G 是连通的，必在回路外面找到一点 v_t 和回路上一点 v_r 相邻接。只要去掉 $v_{r-1}v_r$ 边可得一条从 v_{r-1}到 v_t 的道路:

$$(v_{s-1}, v_{s-2}, \cdots, v_2, v_1, v_p, v_{p+1}, \cdots, v_l, v_{p-1}, \cdots, v_s, v_t)。$$

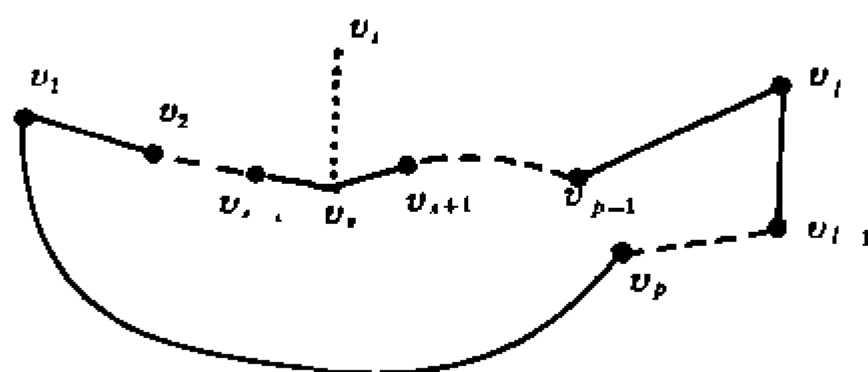


图 1-3-7

如图所示，不难发现这条道路上的边数较前者增多了 1 个(如图 1-3-8 所示)。

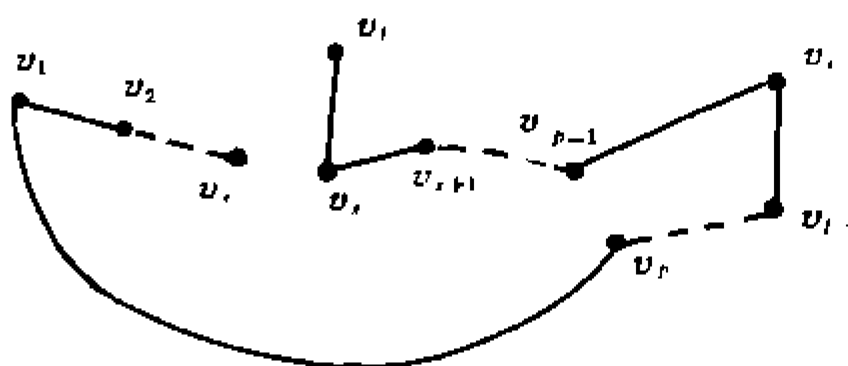


图 1-3-8

不断重复上面的步骤直到存在一条具有 $n-1$ 条边的道路为止。

上面证明所用的构造性方法在图论中常常用到。

例：有 4 个正立方体，如图 1-3-9 所示，它们的 6 个侧面各着以绿、蓝、红、黄 4 种颜色之一。现在要把这 4 块正方体堆成方形柱体，使得这 4 种颜色在柱体的 4 个侧面上各出现一次。

一个正方体有 6 个面，所以 4 个正方体可以堆砌出为数十分可观的不同状态。就是确定了4个正方体依 Ⅰ，Ⅱ，Ⅲ，Ⅳ 次序从上面下排列，只考虑两两接触面不同，就有6^4－

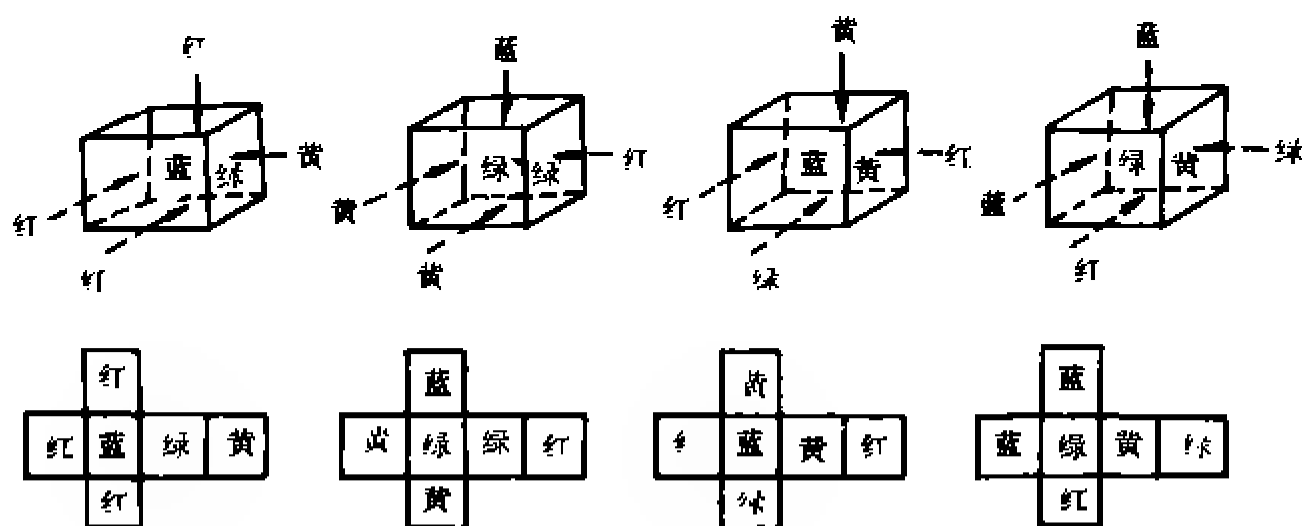


图 1-3-9

1296 种排列，这里还没有考虑 4 个侧面的不同组合。若考虑到后者，又衍生了许多各异的形式来，先令第 I 个正方体保持不动，Ⅱ，Ⅲ，Ⅳ正方体各有 4 个侧面，故有 $4^3=64$ 种状态。因此确定了从上到下的正方体的顺序后，仍然有 $6^4\times4^3=1296\times64=82944$ 种状态。若用穷举法求这问题的解，将是不胜其繁的。

由于上述原因，我们必须寻找解决问题的更好途径。为此，用 4 个顶点 b,g,r,y 分别表示兰、绿、红、黄 4 种颜色。4 个正立方体的各三对面依各对面的颜色连以边，并分别标以 e_1,e_2,e_3,e_4，比如第一个正立方体有一对面着兰、黄两色，则从顶点 b 到 y 引一边标以 e_1，另两对面为红对红和红对绿，故联结 r,r 和 r,g。均标以 e_1。则得图 1-3-10。

从图 1-3-10，找到两个 Hamilton 回路如图 1-3-11 所示，它的 4 条边分别是 e_1,e_2,e_3,e_4。每一回路正好表明一对对面的颜色分布，如图 1-3-12 所示，这便给出了问题的解。

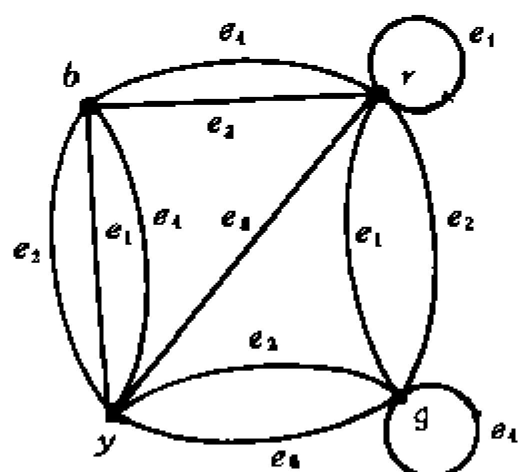


图 1-3-10

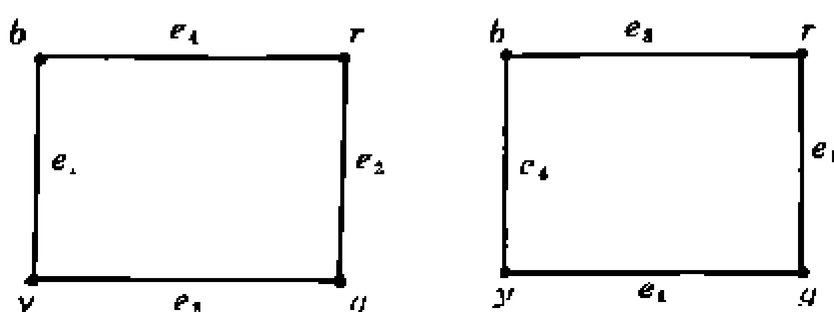


图 1-3-11

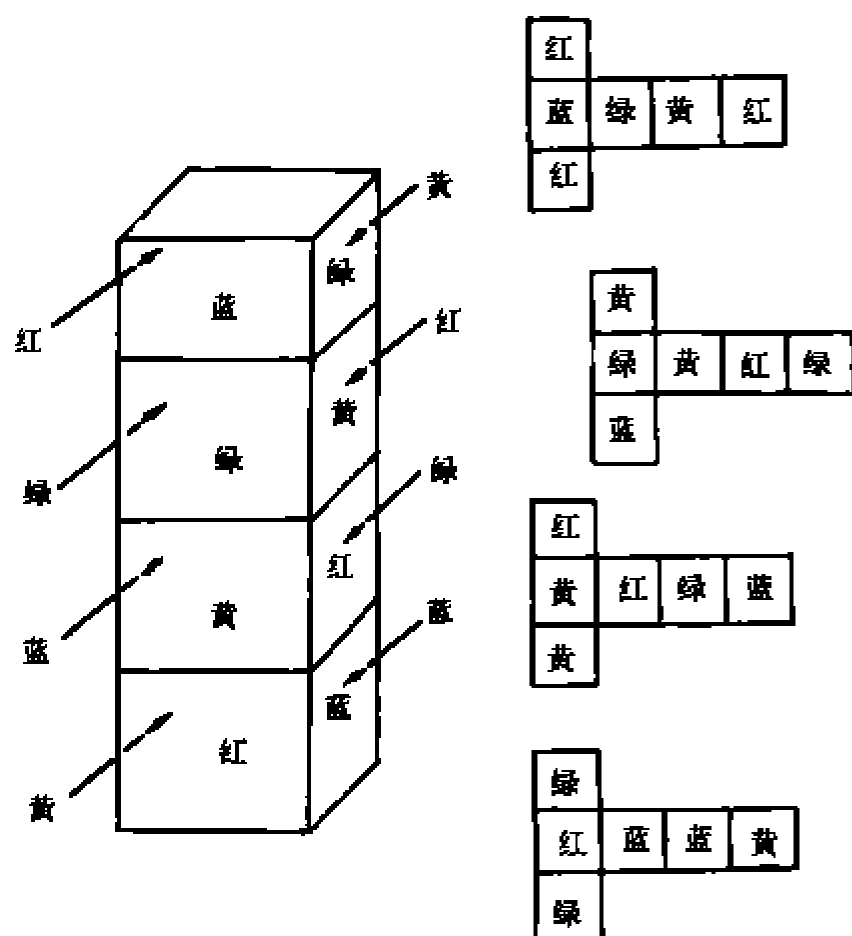


图 1-3-12

§4 图的矩阵表示法

矩阵是研究图论的一种有力工具，特别是利用计算机来研究有关图的算法时，首先遇到的问题是如何使计算机能识别图？利用矩阵识别是其中的一种办法。

以后我们所讨论的图都假定两顶点之间不存在自环和平行的两条边。如不特别说明，一般指无向图。这作为一种约定。

定义 对于图 $G=(V,E)$，构造一矩阵

$$A = (a_{ij})_{n\times n},$$

其中

$$n = V$$

$$a_{ij} = \begin{cases} 1, \langle v_i, v_j\rangle \in E; \\ 0, \text{其它。} \end{cases}$$

则称矩阵 A 是图 G 的邻接矩阵。

给出了图 G 的邻接矩阵，就等于给出了图 G 的全部信息。图的性质就可以从矩阵 A 通过运算而获得。

改变图 G 的顶点排列次序，相当于对邻接矩阵 A 对应的行与列进行调换。以三阶矩阵为例，设

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

若要第二行与第三行互换，相当于左乘以矩阵 P，即

$$PA = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

若要使得矩阵 PA 的第二列与第三列互换，相当于用矩阵 Q 右乘 PA：

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$PAQ = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{13} & a_{12} \\ a_{31} & a_{33} & a_{32} \\ a_{21} & a_{23} & a_{22} \end{pmatrix}$$

一般来说，要使得任意矩阵 $A=(a_{ij})_{n\times n}$ 的第 p 行与第 q 行互换，可左乘以 P 矩阵，它的特点是第 p 行的第 q 列元素为1，而第 q 行的第 p 列为1，其余各行仅主对角线上元素为1，其余元素均为0。若要第 p 列和第 q 列的元素互换，则右乘以矩阵 Q，Q 的第 p 列第 q 行元素为1，第 q 列的第 p 行元素为1，其它各列的主对角元素为1，其余元素均为0。这样

矩阵P每行每列均有一元素而且仅有一元素为1，其余元素为0。

即

$$P=\begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ & \ddots & \vdots & & & & \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ & & \vdots & \ddots & & & \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ & & & & & \ddots & \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \begin{matrix} \\ \\ \leftarrow \text{第}p\text{行} \\ \\ \leftarrow \text{第}q\text{行} \\ \\ \\ \end{matrix}$$

第p列　第q列

$$Q\begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ & \ddots & & & & & \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ & & & \ddots & & & \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ & & & & & \ddots & \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix} \begin{matrix} \\ \\ \leftarrow \text{第}p\text{行} \\ \\ \leftarrow \text{第}q\text{行} \\ \\ \\ \end{matrix} = P^T = P$$

第p列　第q列

图 1.1

这样的矩阵叫做置换矩阵。

从图G的邻接矩阵A可获得图的一些性质，如

(1) $A=(a_{ij})_{n\times n}$中，第i行中非零元素的数目等于顶点v_i的度。

(2) $B=A^2$

$$B=A^2=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = (b_{ij})_{n\times n},$$

其中$b_{ij}=\sum_{k=1}^{n} a_{ik}a_{kj}$。

故b_{ij}表示从v_i两步到达v_j的道路数目。对于有向图G也可类似定义：

$$A=(a_{ij})_{n\times n}$$

$$a_{ij}=\begin{cases} 1, & (v_i,v_j)\in E; \\ 0, & \text{其它。} \end{cases}$$

对有向图的邻接矩阵还有如下性质：

(3) $C=AA^T$

$$C=(c_{ij})=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix},$$

其中 $$c_{ij}=\sum_{k=1}^{n}a_{ik}a_{jk}。$$

故 c_{ij} 的值表示分别以 v_i,v_j 为始点而有共同终点 v_k 的终点数目。如图 1 4 2(a)所示，

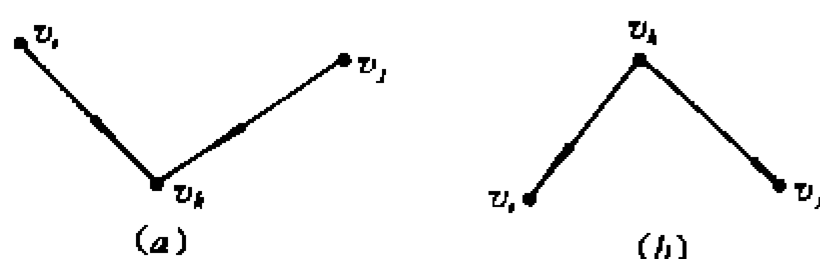


图 1-4-2

(4) $D=A^TA$

$$D=(d_{ij})=\begin{bmatrix}a_{11}&a_{21}&\cdots&a_{n1}\\a_{12}&a_{22}&\cdots&a_{n2}\\\cdots&\cdots&\cdots&\cdots\\a_{1n}&a_{2n}&\cdots&a_{nn}\end{bmatrix}\begin{bmatrix}a_{11}&a_{12}&\cdots&a_{1n}\\a_{21}&a_{22}&\cdots&a_{2n}\\\cdots&\cdots&\cdots&\cdots\\a_{n1}&a_{n2}&\cdots&a_{nn}\end{bmatrix},$$

其中 $d_{ij}=\sum_{k=1}^{n}a_{ki}a_{kj}$。

故 d_{ij} 表示分别以 v_i,v_j 为终点，但有相同始点 v_k 的始点个数。[见图 1 4-2(b)]。

下面我们引入图同构的概念：

设 $G_1=(V_1,E_1)$ 和 $G_2=(V_2,E_2)$，是两个无向图。假如这两个图的顶点集 V_1 和 V_2，边集 E_1 和 E_2 之间都分别建立了一一对应，并且满足这样的关系：一图形的两顶点的边对应于另一图对应顶点间的边，则称这两个图是同构的，即对

$$\forall\ u_1\in V_1,\forall\ v_1\in V_1$$

$$\forall\ u_2\in V_2,\forall\ v_2\in V_2,$$

若 $$u_1\leftrightarrow u_2,$$

$$v_1\leftrightarrow v_2,$$

且 $$(u_1,v_1)\in E_1,$$

$$(u_2,v_2)\in E_2,$$

则 $$(u_1,v_1)\leftrightarrow(u_2,v_2)。$$

举例说明如下：

(1) $v_1\leftrightarrow v_a,v_2\leftrightarrow v_b,v_3\leftrightarrow v_c,v_4\leftrightarrow v_d$；

(2) $e_1=(v_1,v_2)\leftrightarrow e_a=(v_a,v_b)$；

$e_2=(v_1,v_4)\leftrightarrow e_b=(v_a,v_d)$；

$e_3=(v_2,v_3)\leftrightarrow e_c=(v_b,v_c)$；

$e_4=(v_3,v_4)\leftrightarrow e_d=(v_c,v_d)$；

$e_5=(v_1,v_3)\leftrightarrow e_f=(v_a,v_c)$；

$e_6=(v_2,v_4)\leftrightarrow e_g=(v_b,v_d)$；

所以 G_1 和 G_2 是同构的。

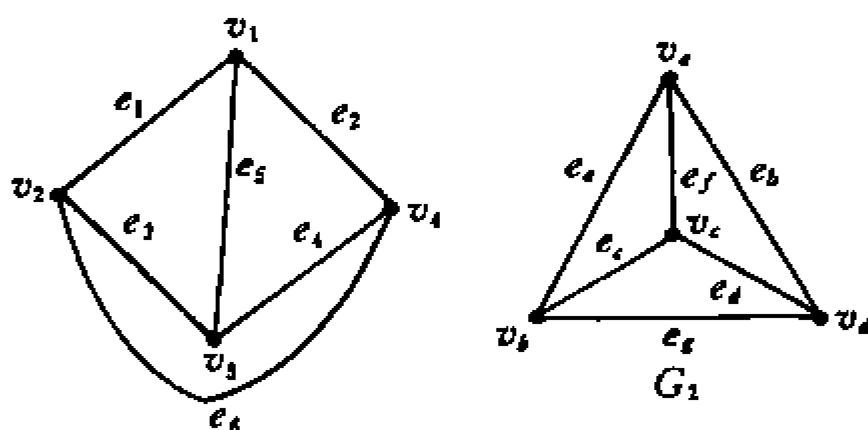


图 1-4-3

所谓图同构，也就是说从外表来看两个图不一致，但它们的拓扑结构是一样的。形象地说，若图的顶点可以任意挪动位置，而边是完全弹性的，只要在不拉断的条件下，这个图可以变形为另一个图，那么这两个图是同构的。

推论 两个图 $G_1=(V_1,E_1)$ 和 $G_2=(V_2,E_2)$ 同构的充分必要条件是存在一置换矩阵 P，使得：

$$A_1 = PA_2P。$$

其中 A_1 是图 G_1 的邻接矩阵，A_2 是 G_2 的邻接矩阵。

如何判定两图同构是图论中的一个困难问题。

§5 中国邮路问题

中国邮路问题，它是由中国数学家管梅谷教授首先提出而得名。设邮递员从邮局出发，遍历他所管辖的每一条街道，将信件送到后返回邮局，要求所走的路径最短。当然如若他所管辖的街道构成一欧拉回路，则这欧拉回路便是所求的路径。如若不然，即存在度数为奇数的顶点时，必然有些街道需走多于 1 遍，如何寻求最短的路？

现将中国邮路问题用图论的语言描述如下：

设 $G=(V,E)$ 是连通图，而且对于所有的 $e\in E$，都赋以权 $c(e)\geq 0$，求从点 $v\in V$ 出发，通过所有边至少一次最后返回 v 点的回路 C，使得

$$\sum_{e\in C} c(e)$$

达到最小。

不失一般性，假定图 $G=(V,E)$ 存在度数为奇数的顶点，如若不然，存在一条欧拉回路，它就是所求的邮路。我们可以设想，有些边添加若干次使得到的图 $G_a=(V,E_a)$ 的所有顶点的度数均为偶数，即 G_a 为欧拉图，问题导致求 G_a 图的欧拉回路，但图 G_a 不再是简单图，它具有平行边，设 e 边重复了 $k(>2)$ 次，去掉偶数条后，仍保持各顶点的度数为偶数，即所得到的图仍是欧拉图。为使总数 $\sum_{e\in C} c(e)$ 达到最小，我们不妨假定每条边重复数目不超过 1。仍使邮路达到最短，也就是要求重复边的长度达到最短。设 E' 是所求的重复边的集合，使 $W(E')=\sum_{e\in E'} c(e)$ 达到最小。对于任一简单回路 C，可分解为与 E' 相交的部分

$E(\bar{C})\cap E^*$，及其余的$E(\bar{C})\setminus E^*$部分。

定理 设$E^*\subset E$是使$W(E^*)=\sum\limits_{e\in E^*}c(e)$达到最小的重复边集合，当且仅当对于$G_a$图的任一回路$\bar{C}$，恒有$W(E(\bar{C})\cap E^*)\leqslant W(E(\bar{C})\setminus E^*)$。

证明 必要性。如若不然，假定存在$\bar{C}$使

$$W(E(\bar{C})\cap E^*)>W(E(\bar{C})\setminus E^*)。$$

这意味着$E(C)\cap E^*$部分的权超过其余的$E(C)\setminus E^*$部分的权，令$\tilde{E}=E(\bar{C})\oplus E^*$，即

$$\tilde{E}=(E(\bar{C})\cup E^*)\setminus(E(\bar{C})\setminus E^*)。$$

$\tilde{E}$也可作为重复边使G图成为欧拉图。这里$\oplus$是对称差，即$0\oplus 0=1\oplus 1=0,1\oplus 0=0\oplus 1=1$。显然，$\tilde{E}$可使图$G$的所有顶点保持其度数为偶数，由于假定

$$\sum_{e\in E(\bar{C})\cap E^*}c(e)>\sum_{e\in E(\bar{C})\setminus E^*}c(e),$$

故

$$\sum_{e\in E^*}c(e)>\sum_{e\in\tilde{E}}c(e)$$

跟E^*的假设相矛盾。必要性得到了证明。

注意这里E^*分解为与$E(\bar{C})$共同部分，还有其余部分，后者出现在$\tilde{E}$中。

充分性证明。假定存在$E^{(i)}\subseteq E,i=1,2$，由$E^{(i)}$的边作为重复的边，满足定理的条件：对于任一回路$\bar{C}$有

$$\sum_{e\in E^{(i)}\cap E(\bar{C})}c(e)-\sum_{e\in E(\bar{C})\setminus E^{(i)}}c(e),\quad i=1,2。$$

可以证明等式

$$\sum_{e\in E^{(1)}}c(e)=\sum_{e\in E^{(2)}}c(e)。$$

令 $\qquad E^*=E^{(1)}\oplus E^{(2)}$，

则图$G^*-(V,E^*)$没有度数为奇数的顶点，E^*可分解成若干简单回路$C_1,C_2,\cdots,C_h$，或记以

$$E^*-E^{(1)}\oplus E^{(2)}=C_1\cup C_2\cup\cdots\cup C_h,$$

则有

$$\sum_{e\in E(C_i)\cap E^{(1)}}c(e)\leqslant\sum_{e\in E(C_i)\setminus E^{(1)}}c(e),i=1,2,\cdots,h。$$

但 $E(C_i)\setminus E^{(1)}=E(C_i)\cap E^{(2)}$，故

$$\sum_{e\in E(C_i)\setminus E^{(1)}}c(e)=\sum_{e\in E^{(2)}\cap E(C_i)}c(e),i=1,2,\cdots,h。$$

同理

$$\sum_{e\in E(C_i)\cap E^{(2)}}c(e)\leqslant\sum_{e\in E(C_i)\setminus E^{(1)}}c(e)=\sum_{e\in E(C_i)\cap E^{(1)}}c(e),i=1,2,\cdots,h。$$

即

$$\sum_{e\in E^{(1)}\cap E(C_i)}c(e)=\sum_{e\in E^{(2)}\cap E(C_i)}c(e),\qquad i-1,2,\cdots,h,$$

所以

$$\sum_{e\in E^{(1)}} c(e) = \sum_{e\in E^{(2)}} c(e)。$$

充分性得证。

从上定理的证明过程提供了构造中国邮路问题的算法。设已知图 G 中顶点的度数为奇数的点为 $v_1, v_2, \cdots, v_{2h}$。

第一步：i 从 1 到 h，引从 v_{2i-1} 到 v_{2i} 的链 P_i，并对 P_i 的每条边附加一条使成为重复边。

第二步：检查图 G 的每条边。若添加的重复边数超过 1，则除去其中偶数条，使得每条边至多有一条添加的边。且每一个顶点的度为偶数，从而得图 G'。图 G' 中重复边的集合记为 $E^{(0)}$。

第二步：$k \leftarrow 0$，

(a) 若 $E^{(k)}$ 对于回路 $\bar{C}$ 有

$$\sum_{e\in E^{(k)}\cap E(C)} c(e) > \sum_{e\in E(C)\setminus E^{(k)}} c(e),$$

则作【$E^{(k+1)} \leftarrow E^{(k)} \oplus E(\bar{C})$，$k \leftarrow k+1$，转(a)】否则转(b)。

(b) 输出 $E^{(k)}$，$E^{(k)}$ 便是最优集。

图 1 5-1 从(a)到(e)给出中国邮路问题求解的过程。(a)便是 G 图，其中 $v_2, v_3, v_4, v_5, v_6, v_7$ 点是度数为奇数的点，引重复边 (v_2, v_3)，(v_4, v_5)，(v_6, v_7) 便得图(b)。(b)中不存在重复边数超过 1 的边。考察回路

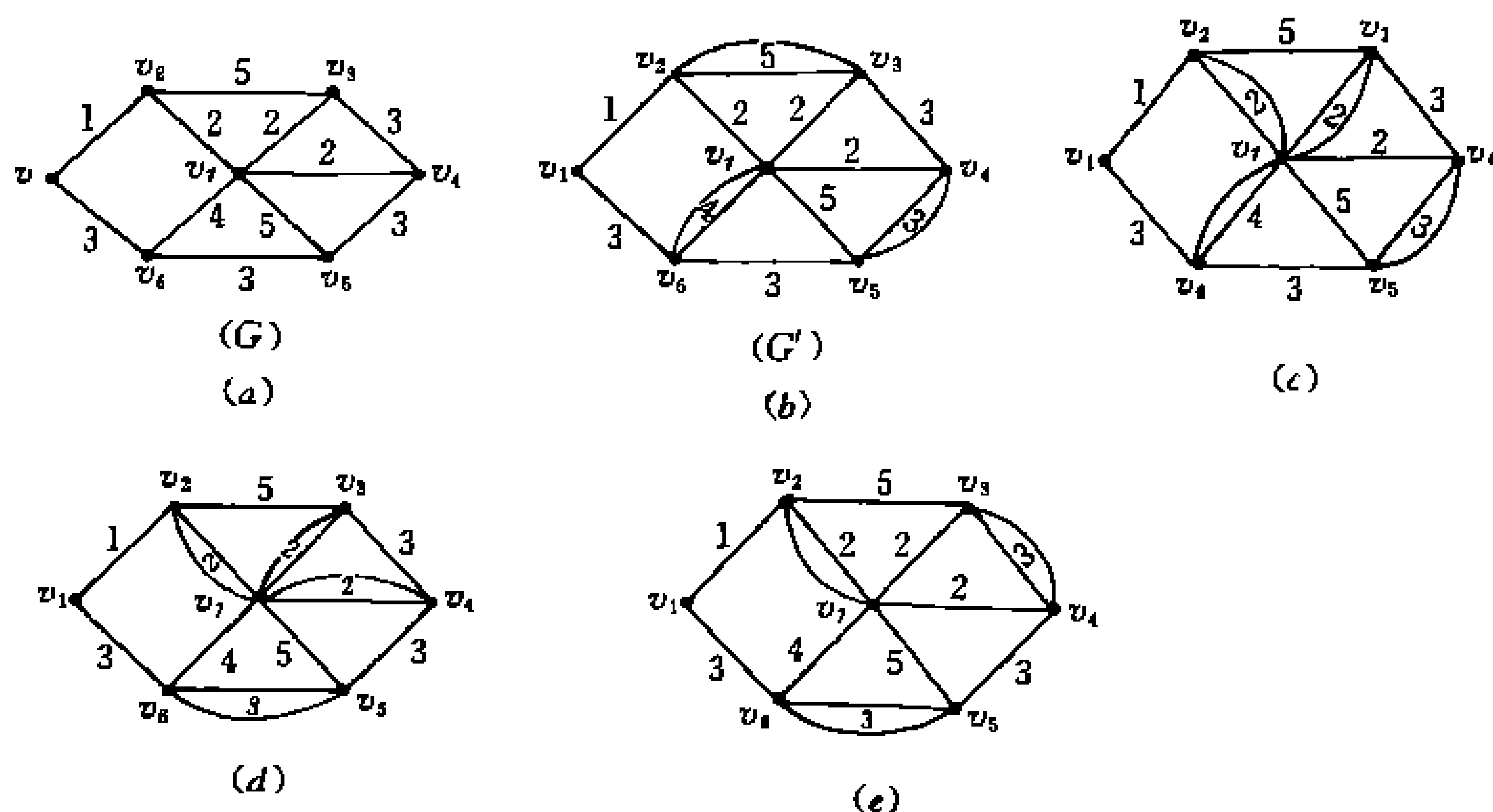


图 1-5-1

$$\bar{C}: v_2 v_3 v_7 v_2,$$

由于

$$\sum_{e\in E'\cap E(C)} c(e) = 5,$$

$$\sum_{e\in E(\bar{C})\setminus E'} c(e) - 2 + 2 = 4。$$

不满足定理的要求，作

$$E^{(1)} = E^{(0)} + E(\bar{C})$$

得图(c)，考察(c)中的回路

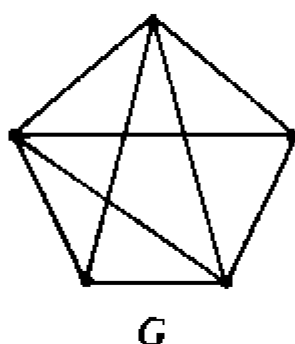
$$\bar{C}: v_4 v_5 v_6 v_7 v_4$$

显然不满足定理的要求，修改(c)得图(d)。图(d)仍不满足定理的要求，再修改得(e)。过程从略。(e)便是前后的结果。

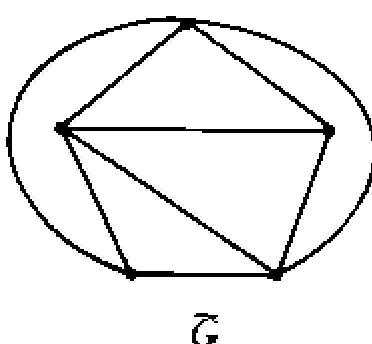
§6 平 面 图

在许多实际问题中，往往涉及到图的平面性的研究，例如单面印刷电路板和集成电路的布线问题，及其某些工程设施等。这章仅讨论平面图的一些最基本的内容。

如果图 G 能够画在曲面 S 上，且除端点处之外任何两条边均不相交，则称图 G 被嵌入曲面 S 上。如果一个图可以被嵌入在平面上，则称为是可平面图，已经被嵌入在平面上的图称为平面图。例如图 1-6-1 所示，G 图可平面化为图 $\tilde{G}$。即图 G 是一个可平面图，$\tilde{G}$ 是 G 的一个平面嵌入，即平面图。



G



$\tilde{G}$

图 1-6-1

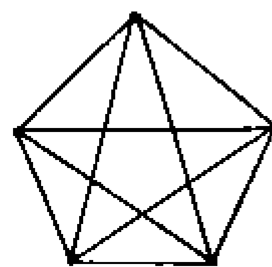


图 1-6-2

不是任何一个图都是可平面图，如图 1-6-2 就不是一个可平面图。

由平面图确定的各个平面域称为面。有界的域称为内部面，无界的域称为外部面。外部面只有一个。

定义 设图 G 是没有环和平行边的平面图。v_i, v_j 是不相邻的任意两顶点，若不能在 v_i, v_j 间增加一条边而不破坏图的平面性时，则称图 G 是最大平面图。

显然最大平面图的每个面都是由三条边包围而成的。例如图 1-6-1 的 $\tilde{G}$ 图便是最大平面图，它不可能再加一条边而不破坏它的平面性了。

在平面图 G 中每增加一条边，边数和面数都增加 1。可通过陆续地增加边使之成为最大平面图。若图 G 的顶点数为 n，边数为 m，域数为 d。增加 p 条边后变成最大平面图 G_1，则 G_1 图的顶点数仍为 n，边数为 $m+p$，域数为 $d+p$，但 $n-(p+m)+(d+p)=n-m+d$ 不变。

欧拉定理 若平面连通图 G 有 n 个顶点，m 条边，d 个域，则

$$n - m + d = 2$$

证明　若图 G 是 n 个顶点的树 T，则 $m=n-1$，$d=1$，$n-m+d=n-(n-1)+1=2$ 成立。每增一条不属于 T 的边，m 增加 1，d 也增加 1，而 $n-m+d=2$ 不变。即

$$(\text{顶点数})-(\text{边数})+(\text{域数})=2$$

不变。证毕。

Euler 公式是平面图的必要条件，不满足这等式的必非平面图。

定理　若图 G 是平面图，不含有平行的边和自环，则图 G 至少有一个顶点的线度小于 6。

证明　设图 G 的顶点数为 n，边数为 m。当 $m=3$ 时有 $d\leqslant 2$。

由于没有平行边和自环，所以每个域的边界至少有三条边，则下列不等式恒成立：

$$3d\leqslant 2m。\tag{1}$$

如果所有的顶点的度都是大于等于 6，故

$$6n\leqslant 2m。(\text{因}\sum\deg(v_i)=2m)\tag{2}$$

由(1)得

$$d\leqslant\frac{2}{3}m。\tag{3}$$

由(2)得

$$n\leqslant\frac{1}{3}m。\tag{4}$$

由 Euler 公式，

$$n+d-m=2,$$

将(3)，(4)代入得：

$$\frac{1}{3}m-m+\frac{2}{3}m\geqslant 2,$$

即

$$0\geqslant 2。$$

导致矛盾。定理得证。

定理　最大平面图的顶点数 n，边数 m，域数 d 满足下列等式：

$$m=3n-6,\qquad d=2n-4。$$

证明　因为已知 G 是最大平面图，

所以
$$3d=2m,$$

以之代入 Euler 公式：

$$d=m-n+2,$$

得

$$\frac{2}{3}m=m-n+2,$$

即
$$m=3n-6。$$

又以之代入 Euler 公式得：

$$d=m-n+2=3n-6-n+2=2n-4。$$

定理完毕。

对于一般平面图当 $d>1$ 时有：

$$m \leqslant 3n-6,\ d \leqslant 2n-4。$$

从上面讨论可知：3 个顶点的图一定是平面图；4 个顶点的图也一定是平面图；5 个顶点的图就不然，比如图 1-6-3 的 $K^{(1)}$，$K^{(2)}$ 便分别是 5 个顶点和 6 个顶点的非平面图的简单例子：

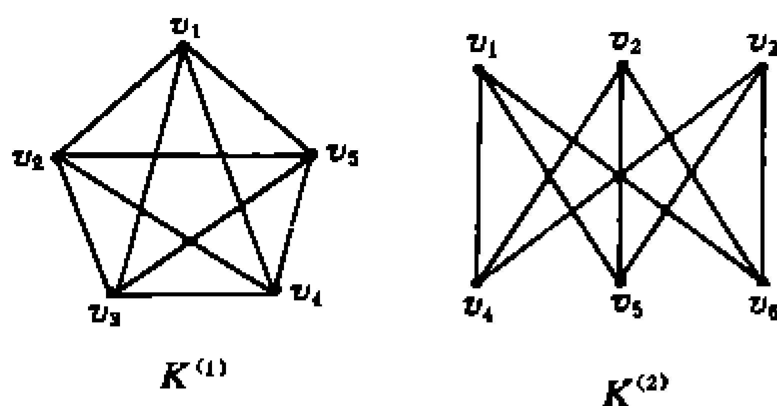


图 1 6 3

这两个图形叫做 Kuratowski 图。

关于 $K^{(2)}$ 图有一个有趣的故事，有 A,B,C 三家住在 v_1,v_2,v_3 三处，彼此交恶不相往来，v_4,v_5,v_6 分别是公用水井、商店和邮局，他们想修道路到 v_4,v_5,v_6 又不在路中碰见，结果办不到。

定理 Kuratowski 图不是平面图。

证明 $K^{(1)}$ 图中的 $n=5$，$m=10$，故它不满足平面图的必要条件

$$m \leqslant 3n-6。$$

至于 $K^{(2)}$，由于 $n=6$，$m=9$，如果 $K^{(2)}$ 是平面图，则

$$d = 2+m-n = 2-6+9=5。$$

但 $K^{(2)}$ 图的任意一封闭曲线至少有 4 条边组成，即 $4d \leqslant 2m$，故导致 $20 \leqslant 18$ 的矛盾。这就证明了 $K^{(2)}$ 不是平面图。

在 Kuratowski 图 $K^{(1)}$，$K^{(2)}$ 的边上加入任意个线度为 2 的顶点所形成的图显然不改变其非平面图性。与这些图同型的图分别叫做 $K^{(1)}$ 型图和 $K^{(2)}$ 型图。$K^{(1)}$ 型图和 $K^{(2)}$ 型图统称为 K 型图。如图 1 6 4 所示。

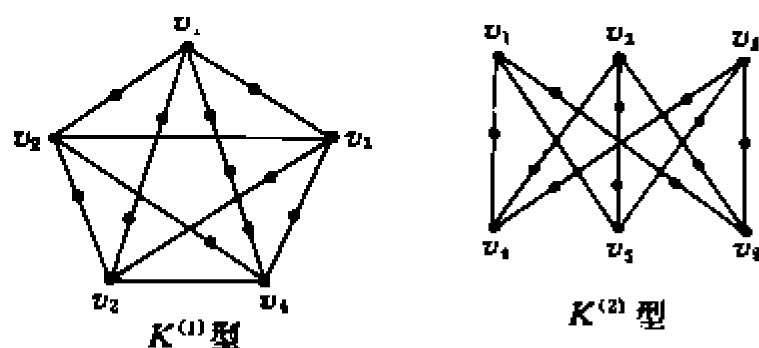


图 1-6-4

Kuratowski 定理 图 G 是平面图的充分必要条件是 G 图不存在任何子图为 $K^{(\ )}$ 型

图或$K^{(2)}$型图。

证明较繁略去。Kuratowski 定理给出了判断可平面图的充要条件，似乎很直观，但实际上真要用于判断比较复杂的图还是非常困难的。

现在我们简单介绍一下对偶图的概念。

设G是一平面图，如果G的两个面的边界至少有一条公共边，则称这两个面是相邻的。

设G是有n个顶点，m条边和d个面的平面图，令G的d个面为$S_1,S_2,\cdots,S_d$。在G的每个面S_i内部放置一个顶点v_i^*，如果面S_i和S_j相邻，则用边(v_i^*,v_j^*)连接v_i^*和v_j^*点使它与面S_i,S_j的公共边只相交一次，且与G的其它边界无交点。这样得到的图G^*称为G的对偶图。

显然图G^*有d个顶点，m条边。

例：

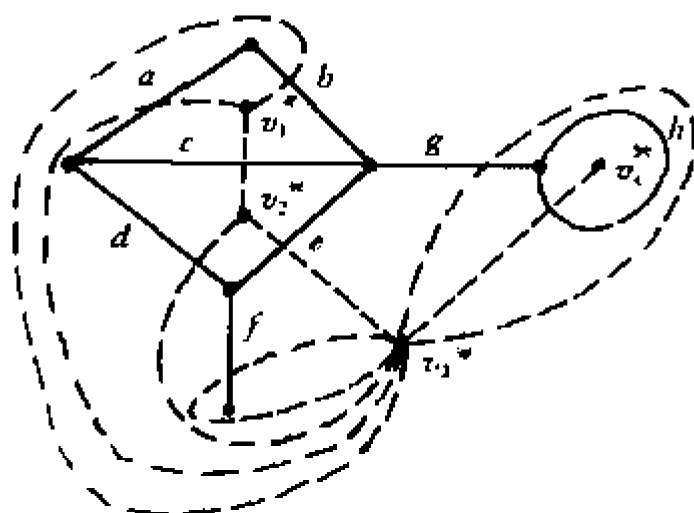


图 1-65

得到图G的对偶图G^*以后，在图G上的许多问题可以化为相应的对偶图的问题，如下面的着色问题也是对偶图的一个应用，由域着色可以转化为点着色，反之亦然。

所谓平面图的着色问题，指的是保证平面图有相邻边的域着不同的颜色，问最少要几种颜色？如地图着色等实践经验说明：在任何情况下有 4 种颜色足够了，但要严格地论证这一点却不那么容易。这就是图论中著名的“四色问题”：也是历史上有名的数学难题。这个问题在 1976 年由 K. Appel 和 W. Haken 两位数学家在计算机的帮助下获得了解决，电子计算机运转了 1200 小时，作了近百亿次判断后给出了结果，是机器证明的一大事件。可惜后来发现有错误。

这里我们证明“五色问题”，即对于任何平面图G只要 5 种颜色足以使得相邻的域有不同的颜色。

五色问题可以化为它的对偶图的顶点着色问题，使得相邻的两顶点没有相同的颜色。证明采用数学归纳法。

不失一般性不妨假设图G是连通的，没有自环和平行的边。

如果G图的顶点个数少于 5，则四色显然足矣。

假设顶点个数等于$n-1$时五色问题的答案是肯定的，进而证明顶点个数为n时也必

然是对的。设图 G 有 n 个顶点。

由于 G 图是平面图，根据平面图性质的定理，顶点中度小于 6 者必然存在，设为 v_0，$d(v_0)\leqslant 5$。

若把 v_0 点以及与 v_0 相连的边一起从 G 图中消除（但边的另一端点保留），令所得的新图为 G_0，则 G 的顶点个数为 $n-1$。根据假设对 G_0 图定理是正确的。

如果 $d(v_0)=4$，则五色显然是够的，因与 v_0 点相邻的顶点只有 4 个，最多用了 4 种颜色，余下一色正好给 v_0。

问题在于若 $d(v_0)=5$ 时，证这时五色依然是对的。

设 v_0 的 5 个相邻顶点 v_1,v_2,v_3,v_4,v_5，按逆时针方向依次排列如图 1-6-6 所示，分别着以 a,b,c,d,e 5 种颜色。

令 G_0 图中着以 a,c 两色的顶点及其间连线构成 G_0 的子图，用 G_{ac}表示。显然 v_1,v_3 两点属于 G_{ac}图。依据分别着 a,c 色的 v_1,v_3 两点在 G_{ac}图中连通与否分别讨论如下：

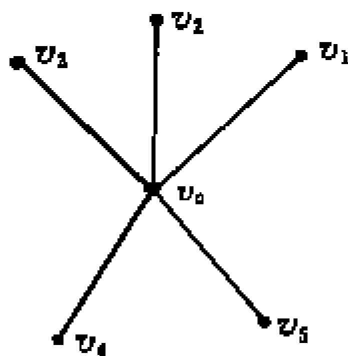


图 1-6-6

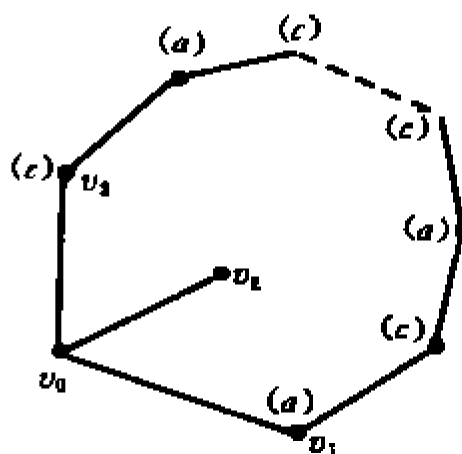


图 1-6-7

(1) v_1,v_3 在 G_{ac}中不相连。即 v_1,v_3 分别属于不相连的两个连通子图中。若令含 v_1 点的连通部分 a,c 两色互换一下，即 v_1 点改为着 c 色，不至于影响扩大到 v_3 点，其结果使得 v_1,v_3 具有相同的颜色 c。颜色 a 留给 v_0 点。五色定理得证。

(2) 若 v_1,v_3 两点在 G_{ac}中相连，则从 v_1 到 v_3 有一条道路，沿着这条道路，a 色，c 色交替出现，如图 1-6-7 所示。(a)表示着色 a，(c)当然是着 c 色。

这条道加上(v_3,v_0)边和(v_0,v_1)边形成一回路。这时在由顶点着 b 或 d 色构成的 G_{bd} 子图中分别着 b 和 d 色的 v_2 和 v_4 点显然不属于相连通的部分中，这样，令 G_{bd}子图中 v_2 所在的连通部分 b,d 二色互换，不至于影响到 v_4 点着色。故给 v_0 点着以 b 色即可。定理证毕。

§7 Petri 网

前面我们讨论了图论的基本概念和几个实际的问题，从中可以看出，利用图来描绘各种实际问题取得了形象直观的效果，许多看起来非常麻烦的问题用图来表示变成一目了然。但那都是非常古老的话题。近年来又有一些进展，Petri 网就是其中突出的一例。下面简单介绍一下这一新的领域。

Petri 网是研究复杂系统模型的一种数学工具，它通过对实际问题构造 Petri 网并对 Petri 网进行分析，从而揭示出系统的动态特性和重要信息。1962 年它是由 Petri 在其博士论文中首先提出的，由于它有着广泛的应用前景，所以一直受到重视。如果说以前谈到的图论方法用来表示静态的关系，显示了它的作用，对动态的现象它便显得无力，Petri 网弥补了它的不足。

一、Petri 网的概念

Petri 网包含以下部分：

1. 位置集合 P

$P=\{p_1,p_2,\cdots,p_n\}$是位置点的有限集合，$n\geqslant 0$；

2. 转移集合 T

$T-\{t_1,t_2,\cdots,t_m\}$是转移点的有限集合，$m\geqslant 0$；

P 和 T 无公共部分，即交集 $P\cap T$ 为空集。

3. 输入功能 I

$I(t_i)$为 P 的子集，即有有向边指向 t_i 的点。若 $I(t_i)=\{p_{j_1},p_{j_2},\cdots,p_{j_k}\}$，则存在有向边 (p_{j_h},t_i)，$h=1,2,\cdots,k$。不要求 p_j 互不相同；

4. 输出功能 O

$O(t_i)$也是 P 的子集，即有有向边由 t_i 发出指向的点，若 $O(t_i)-\{p_{k_1},p_{k_2},\cdots,p_{k_r}\}$，则存在有向边$(t_i,p_{k_l})$，$l=1,2,\cdots,r$。

我们将此 Petri 网记作 $C-(P,T,I,O)$。

实际上 Petri 网可看成一个有向图 $G-(V,E)$，其中

$$V=P\cup T;$$
$$E-I\cup O。$$

从这个意义上说 Petri 网也是图，不过是一种很特殊的图。顶点区分为 P 和 T 两种不同类型。

例：$C-(P,T,I,O)$；

$P=\{p_1,p_2,p_3,p_4,p_5,p_6\}$；

$T=\{t_1,t_2,t_3,t_4,t_5\}$。

I：

	t_1	t_2	t_3	t_4	t_5
I	$\{p_1\}$	$\{p_3\}$	$\{p_2,p_1\}$	$\{p_4,p_5,p_5,p_5\}$	$\{p_2\}$

，

O：

	t_1	t_2	t_5	t_4	t_5
O	$\{p_2,p_3\}$	$\{p_3,p_5,p_5\}$	$\{p_2,p_4\}$	$\{p_4\}$	$\{p_6\}$

。

即 Petri 网是一有向图，它的顶点分属于 P 和 T 的两类，为加以区别，属于 P 的点用“○”表示，属于 T 的点用“-”表示它，边用输入输出功能来确定。例如对于上例有图 1-7 1。比如 $I(t_4)=\{p_4,p_5,p_5,p_5\}$，故引向 t_4 的有一有向边来自 p_4，3 条有向边来自 p_5。

利用 Petri 网还可表示并行算法。例如求 $ab+cd$ 可用图 1-7 2 来表示：即 $a*b$ 和 $c*$

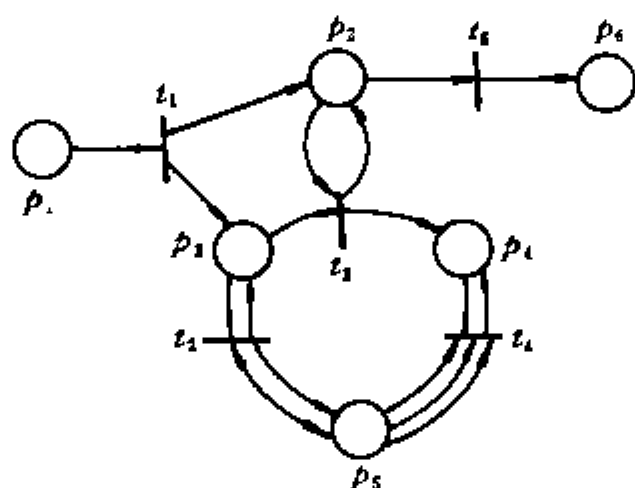


图 1 7 1

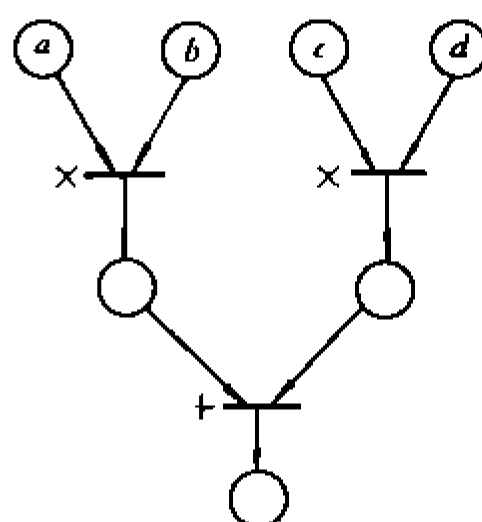


图 1-7-2

d 可在同一时刻进行计算。

二、对偶 Petri 网

Petri 网 $C=(P,T,I,O)$的对偶 Petri 网是由 C 中位置集 P 和转移 T 互换而得到的，它的 Petri 图是 C 的 Petri 图中"○"改为"|"，" "改为"○"而成的图。对偶的 Petri 网 $\bar{C}=(P,T,I,O)$也是 Petri 网。例如，图 1-7-3 是图 1 7 1 的对偶图。

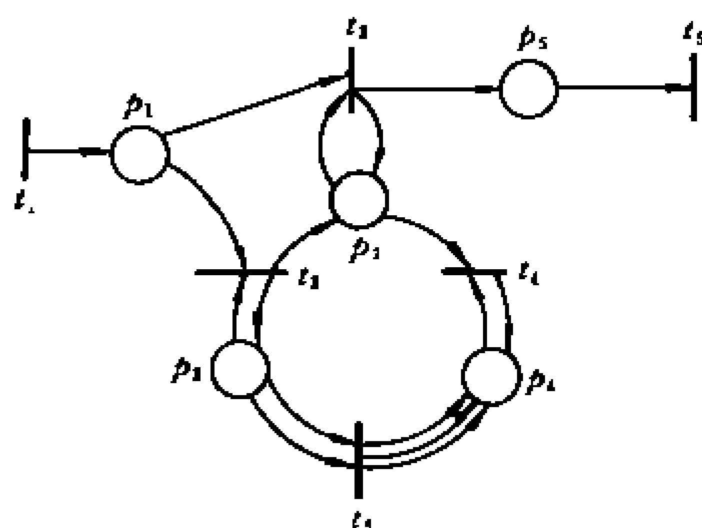


图 1-7-3

三、带标志的 Petri 图

带标志的 Petri 网是给 Petri 网的位置点"○"以标志"·"，每 -位置点"○"给的标志"·"的数目可以不限，Petri 网的结构相同，但标志数目可以不同。网络结构虽然一样但标志"·"的数目不同，表达了不同的两种状态。

该 Petri 网的性能可以直观地叙述如下：若 $p\in I(t)$，且 p 点的标志"·"数目不少于由 p 到 t 的边的数目，则 t 将被启动，或称 t 可以点火。t 点火后，t 的输入位置的点 的标志"·"数目减少，减少的数目等于(p,t)边的边数，t 的输出位置点 p' 的标志"·"数增加，增加的数目正好是(t,p')边的边数。即通过转移点 t，将标志"·"从输入位置 p 转到输出位

置 p'，一个转移点点火只当它的每一个输入位置的标志"·"数至少和它通向转移点的边数一样多。若将位置点看作是条件，而转移点是执行或运算，则标志数则用以刻划条件是否成熟。例：

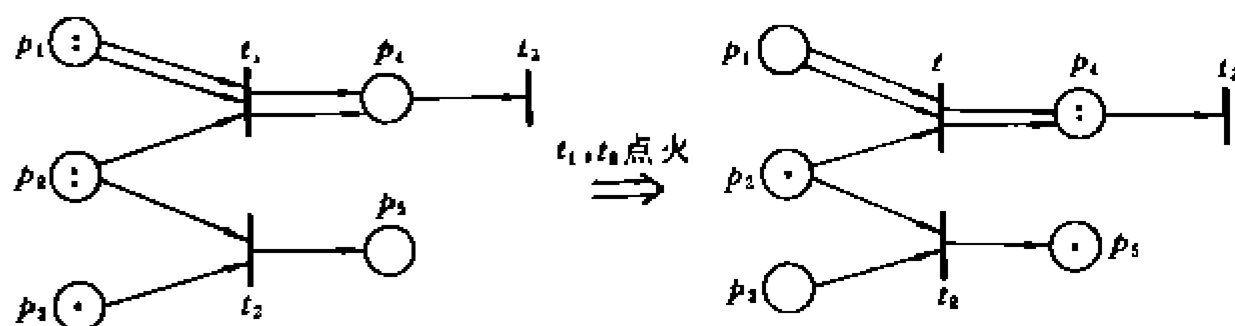


图 1-7-4

Petri 网的状态由它的标志所确定，每一转移点的点火将改变它的状态。还是那句话虽然 Petri 网一样，但其标志不同，所表示的状态就不同。

下面举例说明如何构造复杂系统的 Petri 网。

例 1：设有三台机器 m_1,m_2,m_3，有两个操作人员 A 和 B，A 能操作 m_1 和 m_2，B 能操作 m_1 和 m_3，加工顺序：第一阶段在 m_1 上加工，第二阶段在 m_2 或 m_3 上加工，加工开始必须具备(1)有任务在等待；(2)机器空闲。

问题的 Petri 网如图 1-7-5 所示，其中：t_1：任务等待加工；t_2：A 操作 m_1；t_3：操作 m_1 完毕；t_4：B 操作 m_1；t_5：B 操作 m_1 完毕；t_6：A 操作 m_2；t_7：A 结束 m_2 的操作；t_8：B 操作 m_3；t_9：B 结束 m_3 的操作；t_{10}：将加工完毕的任务送走。

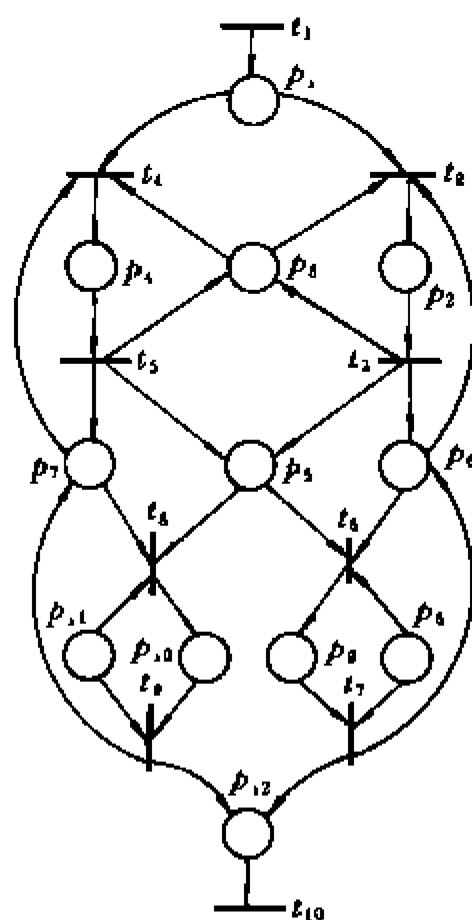


图 1-7-5

例 2：某问题的算法如下：

S_1：$y_1 \leftarrow a$，$y_2 \leftarrow b$，$y_3 \leftarrow 1$；

S_2：若 $y_1>0$ 转 S_3，否则转 S_5；

S_3：若 y_1 是偶数则转 S_4，否则作：

$y_3 \leftarrow y_3 * y_2$；$y_1 \leftarrow y_1 - 1$；

S_4：$y_2 \leftarrow y_2^2$，$y_1 \leftarrow y_1 - 2$；转 S_2；

S_5：输出 y_3，结束。

上述算法可用流程图 1-7-6 表示。它的 Petri 网图见图 1-7-7。

其中 t_1：完成 S_1 运算；

t_2：$y_1>0$；

t_3：y_1 是奇数；

t_4：y_1 是偶数：

t_5：完成 $y_3 \leftarrow y_3 * y_2$，$y_1 \leftarrow y_1 - 1$；

t_6：$y_2 \leftarrow y_2^2$，$y_1 \leftarrow y_1 - 2$；

t_7：$y_1 \leqslant 0$；

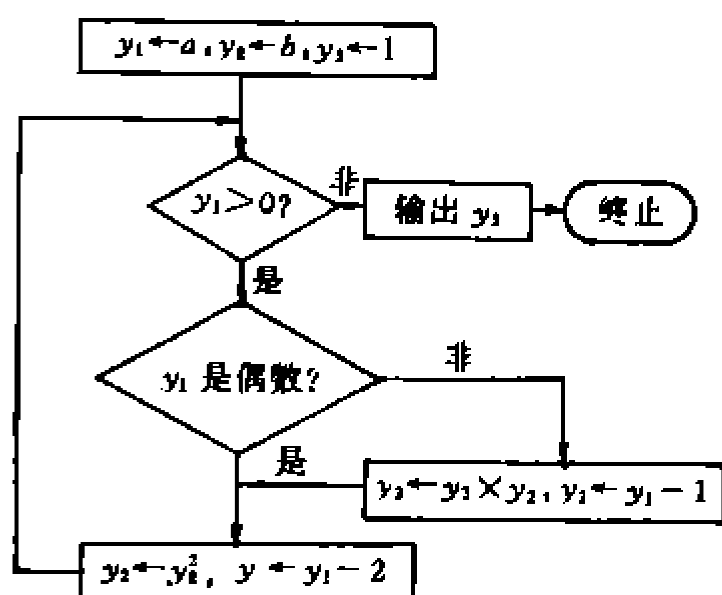


图 1-7-6

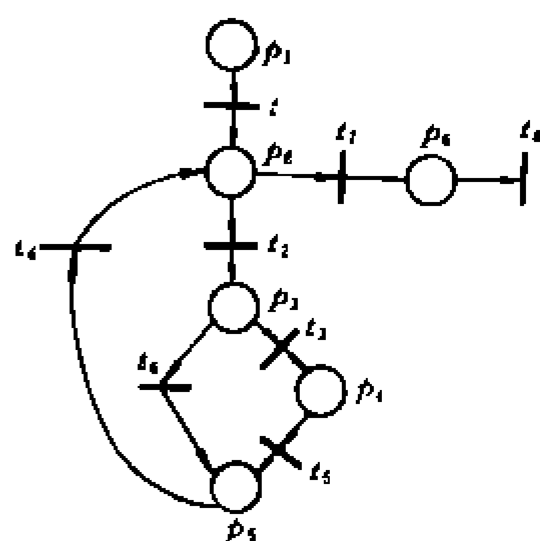


图 1 7 7

t_8：输出 y_3。

例 3：如图 1 7-8 所示的操作系统。

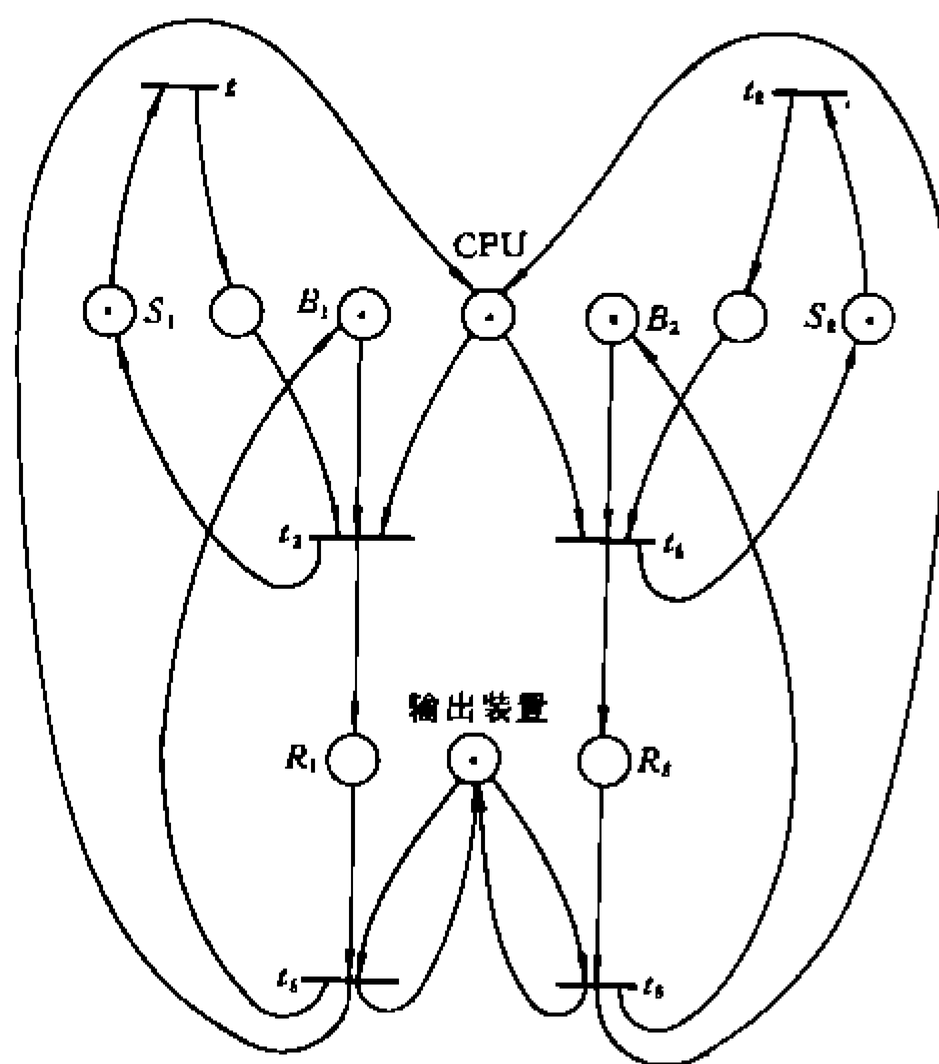


图 1 7 8

其中 B_1 和 B_2 是两个存储区，其它的位置点都是资源，“・”表示该资源有效，转移表示操作系统的操作，开始时CPU 有一个“・”，标志它处在有效状态，任务的最后结果通过输出装置输出，t_1 和 t_2 是两个终端。

但图 1 7-8 所示的操作系统的模型允许出现 t_1，t_3 交替点火，为此可将其改进为图 1

7-9 所示的 Petri 网。

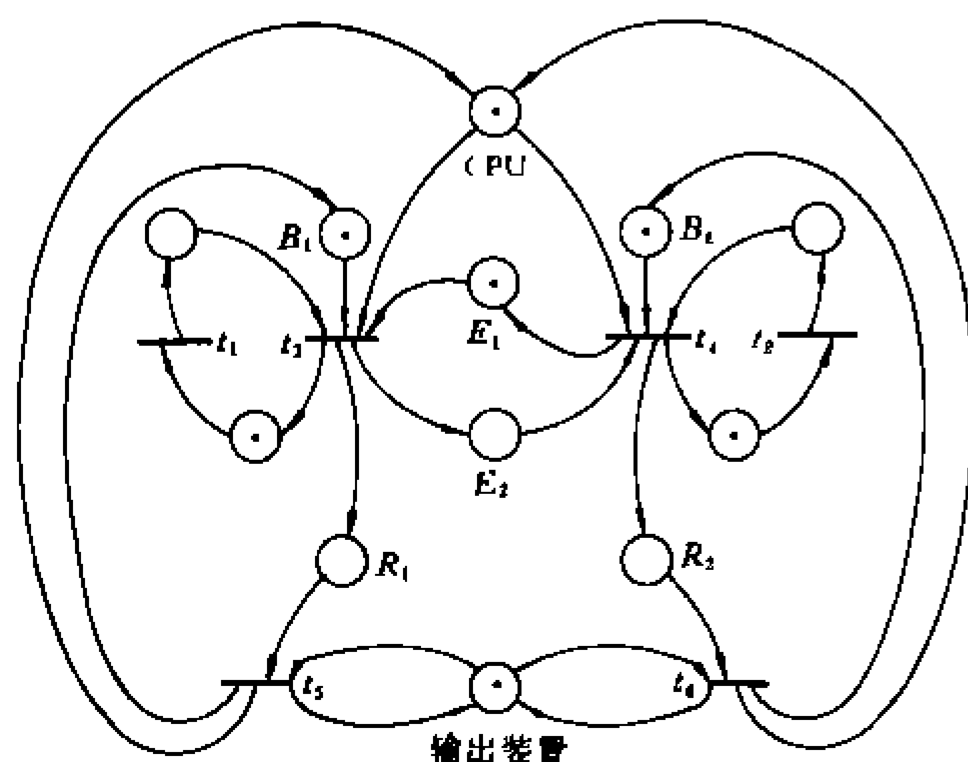


图 1-7-9

这样 t_3 点火后，E_2 有一"。"，避免 t_3 再点火，直到 t_4 点火后，E_1 获得一个"。"，t_3 才获得点火的机会。

例 4：5 个哲学家问题。

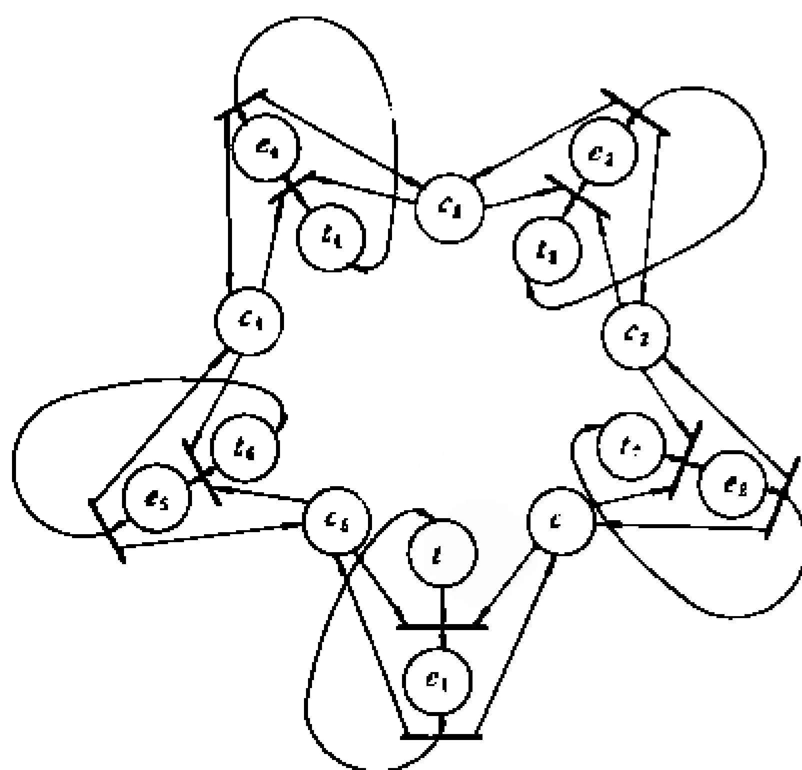


图 1 7 10

5 个哲学家围着圆桌坐下，每两个哲学家之间都放着一根筷子。吃中国菜要一双筷子，所以每位哲学家要吃中国菜必须取他左右的 一双筷子．如果每位都取他左边一根等待另 一根，则将陷入死锁。下面用 Petri 网叙述问题的解。设 c_1,c_2,c_3,c_4,c_5 为 5 根筷子，e_i 和 t_i 分别表示第 i 个哲学家吃饭和想问题两种状态，$i=1,2,\cdots,5$，当哲学家从思索状态转入

吃的状态，则他必须有左右两根筷子。

习　题

1. 某甲挑一担青菜、赶一条狗、一只羊赶路，途经一小河，船小一次只容许某甲带狗、羊、菜三者其中的一种过河。当某甲不在场时，应避免狗和羊或羊和菜在一起，求过河的方案。
2. 有三个瓶子，容量分别为 8，5，3 升。若容量为 8 升的瓶子装满了液体，试利用这三个瓶子把 8 升的液体均分为 2。
3. 若 §1 的(三)中若附加一条件：不允许 2 敌人单独呆在一起，求渡河的方案。
4. 若 §1 的(三)中，敌我双方人员各为 3 人，求渡河方案。
5. 山上一生产队要用拖拉机运两头牛、三只羊下山。拖拉机下山时只能运载两只牲口，上山时只能运一只。人不在场时若牛、羊在一起，则要求羊的头数超过牛的头数。求运输方案。
6. 试证任意 10 人中必有 3 个人互相认识或有 4 个人互不相识，两者必居其一。
7. 图 G 有 n 个顶点、m 条边，若：

$$m > \frac{1}{2}(n-1)(n-2)$$

试证图 G 必是连通图。

8. §3 中的 4 个正方体能否堆成一方形柱体使得每个侧面都是同一颜色？为什么？
9. n 个人出席一圆桌会议，会议主持者希望每次每人坐位两旁的人都和以前的不同，问这样的坐位方案最多有几种？n 为奇数时试设计一组这样的方案。
10. 设图 G 的顶点数为 n，对于图 G 的任意一对顶点 v_i，v_j，恒有：

$$d(v_i) + d(v_j) \geqslant n$$

试证图 G 必至少存在一条 Hamilton 回路。

11. 试证下列两图同构，

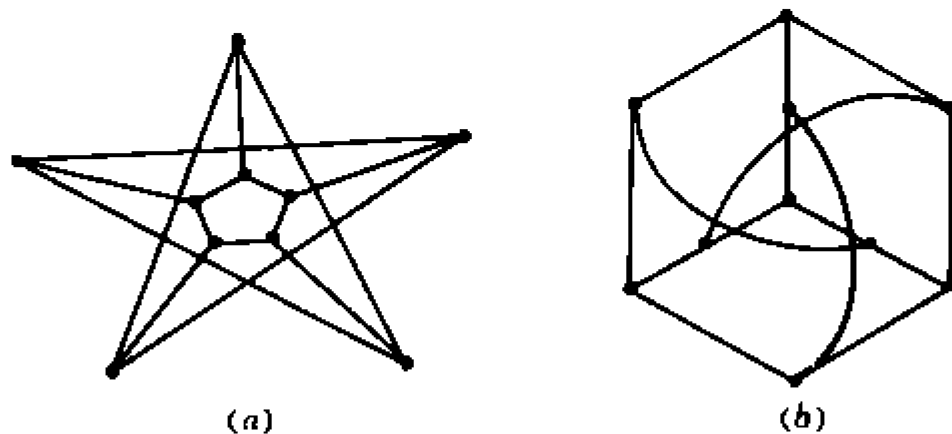


图　习题 11

12. 有 n 个单位各派代表 2 人组成的代表团，分到这 n 个单位去监查工作，每单位派去两人。派去该单位的代表要求与所去的单位平时无工作联系。现假定这 n 个单位中每一

个与其它单位平时有工作关系的少于$\left[\frac{n-1}{2}\right]$，试问应如何设计，并证其方案是存在的。

13. 一计算机系统有一台打印机 P 和一台磁盘驱动器 D。两位用户共享这个系统，而且都对 P 和 D 提出请求，试用 Petri 网描述这系统的模型，并讨论之。

14. 用 Petri 网表示下面的程序

(1) $x \leftarrow 1$，
$y \leftarrow 2$
$z \leftarrow x+y$
$z \leftarrow z+z$

(2) $x \leftarrow 2$
$y \leftarrow x+x$
$z \leftarrow 2$
$w \leftarrow x+x$
$z \leftarrow x+y+z$

15. 试判定下列图形两两同构。

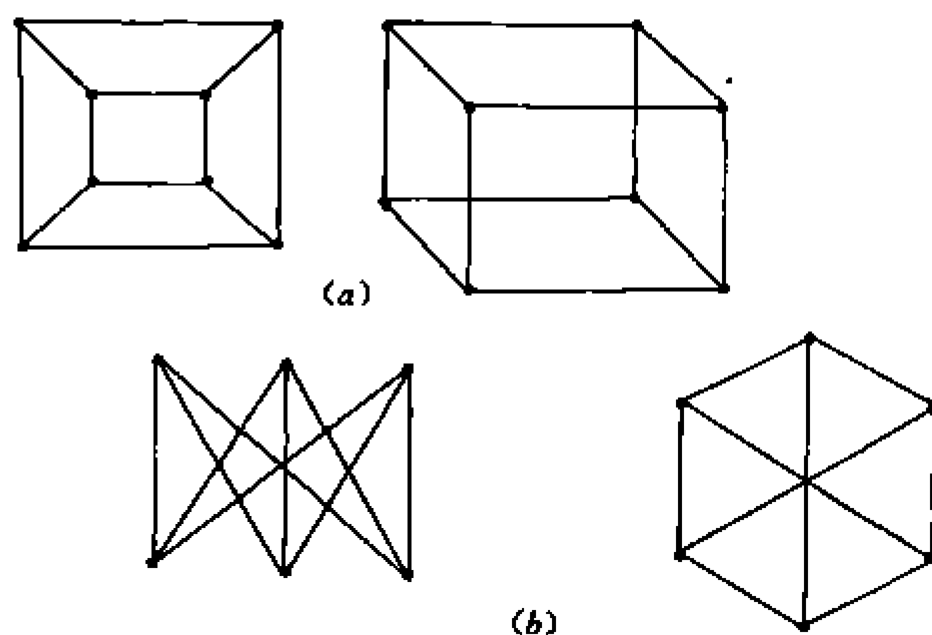


图 习题 15

16. 设图 G 是一 n 个顶点的简单图，图 G 的补图 $\bar{G}$ 定义如下，G 和 $\bar{G}$ 有相同的顶点集合 V，若 u 和 v 在 $\bar{G}$ 相邻，则在 G 不相邻，反之亦然。试求下列图形的补图。

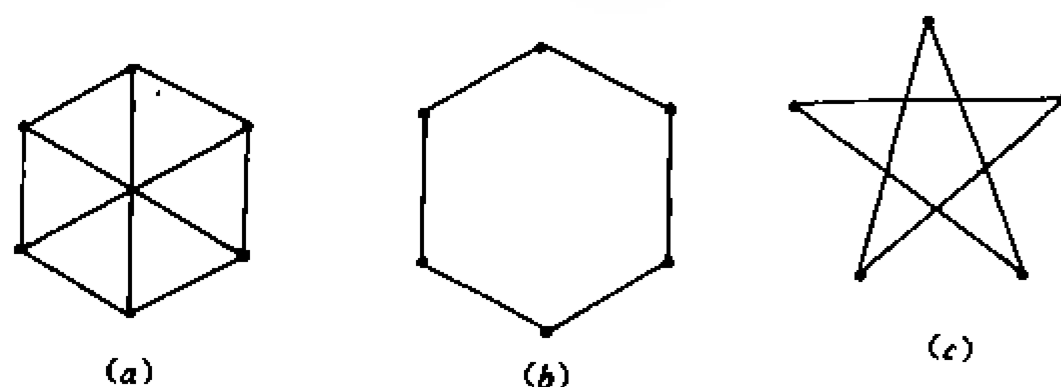


图 习题 16

17. 求一条回路过 Königsberg 七座桥，要求重复的次数尽量少。

18. 试作 $n=2$ 的 de Bruijn 图。

第二章　树

树是图论中最重要的概念之一，它是基尔霍夫在解决电路理论中求解联立方程问题时首先提出的。它又是图论中结构最简单，用途最广泛的一种连通图。本章讨论树的概念、性质，主要是某些应用。

§1　树的概念

我们首先从一个问题谈起。图 2-1-1 是通讯线路图。其中 $v_1, v_2, \cdots, v_{10}$ 是 10 个城市，线路只能在这里相接。不难发现，只要破坏了 (v_7, v_9) 和 (v_4, v_5) 两条线路，就使这个通讯系统分解成互不相连的两部分。但是要问在什么情况下这十个城市依然保持彼此相通？那情况要复杂多了。不难想象最少要有 9 条线把这 10 个城市连接在一起。而且这 9 条线路是不存在任何回路的。因而 9 条线路少了一条都将使得系统失去连通性。

为了解决以上一类的问题，我们引入树的概念。

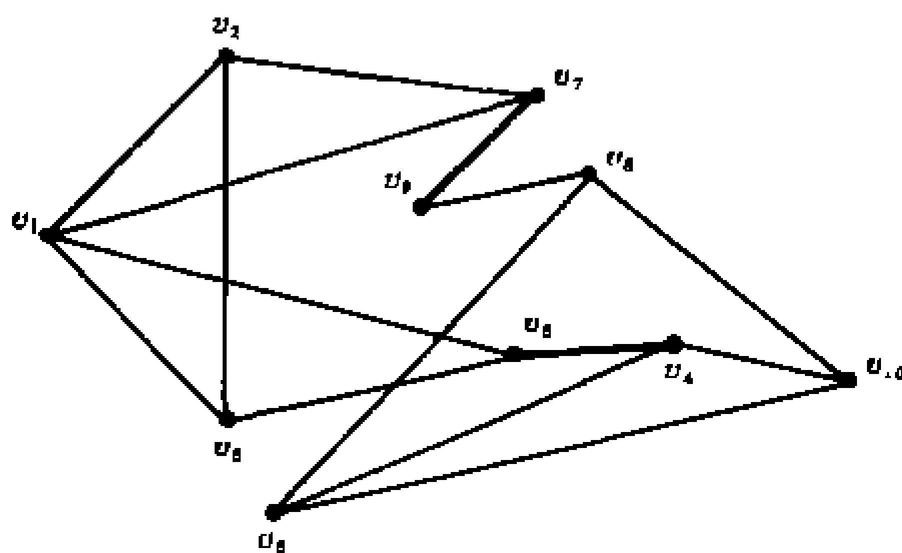


图　2-1-1

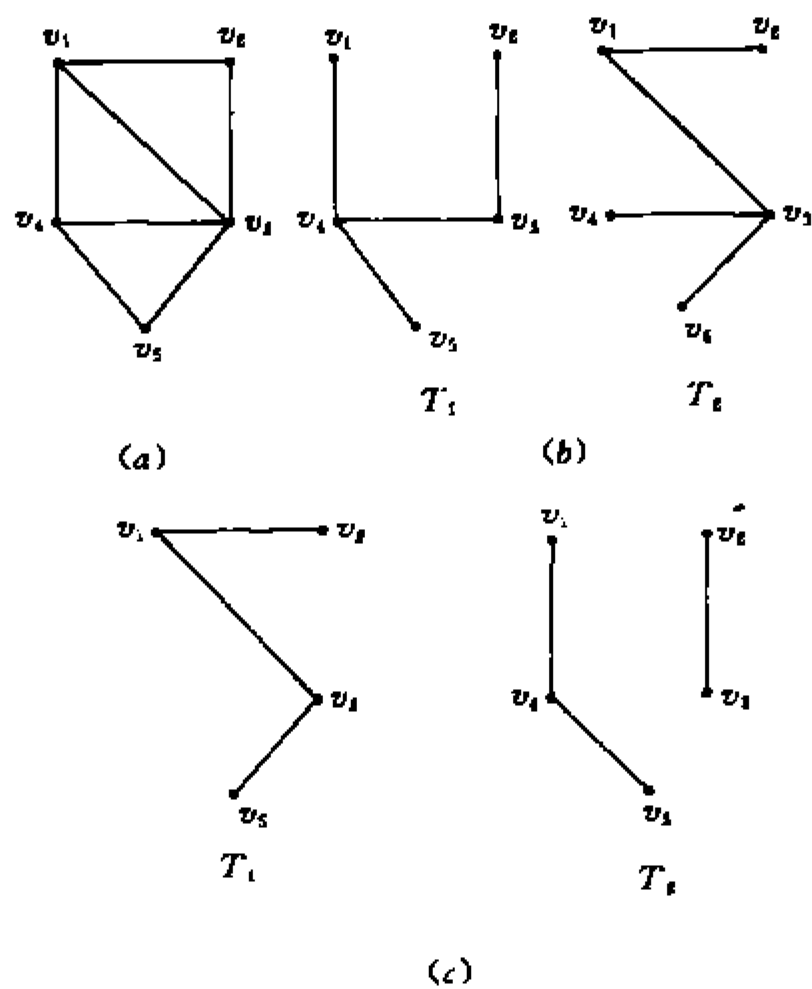
没有回路的图称为无圈图。连通的无圈图称为树。图 G 的连通无圈子图称为 G 的树。若图 G 的树包含图 G 的所有顶点，则称为图 G 的一棵生成树或支撑树。树 T 的一个连通子图称为 T 的子树。例如图 2-1-2 所示，(a) 表示图 G，(b) 中 T_1，T_2 分别是图 G 的生成树。

T 是图 G 的树，从图 G 中去掉树 T 的边后所得的子图称为树 T 的余树，记作 $\overline{T}$。如图 2-1-2(c) 中 $\overline{T}_1$、$\overline{T}_2$ 分别是(b)中 T_1 和 T_2 的余树。属于树 T 的边称为树枝；$\overline{T}$ 中的边称为图 G 的弦。由于树 T 唯一确定余树 $\overline{T}$，故也可称其为树 T 的弦。由上可知道余树 $\overline{T}$ 不一定连通。

下面再来看一下树在其它科学领域的一些应用：

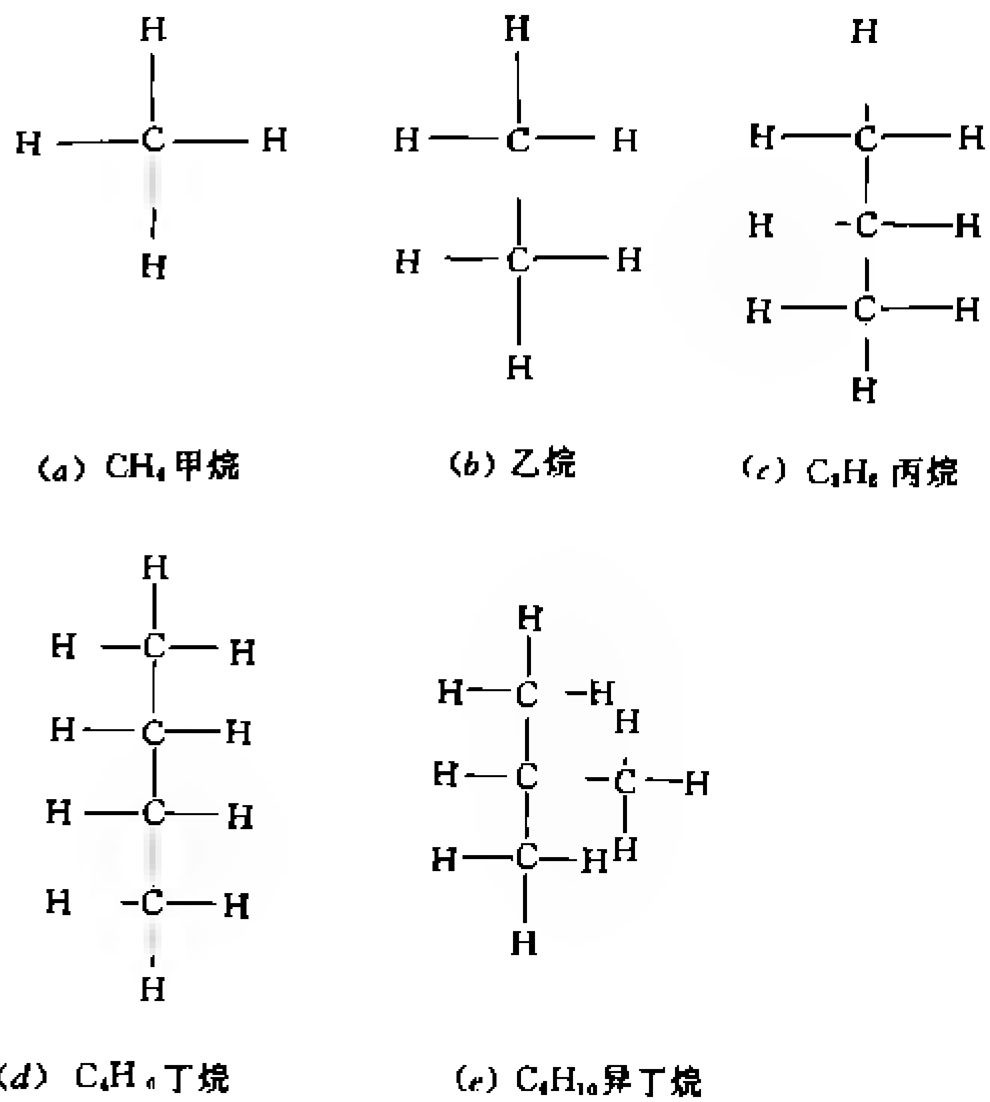
例 1：有机化学中碳氢化合物 C_nH_{2n+2} 随着 n 取不同的整数而为不同的化合物。例如 $n=1$ 时为甲烷；$n=2$ 时为乙烷；$n=3$ 时为丙烷；$n=4$ 时则由于化学枝链结构不同而有丁烷、异丁烷之分。它们的化学链分别如图 2-1-3 所示。

一般说来，枝链的结构规律是碳原子 C 的度为 4，氢原子的度为 1。对于 C_nH_{2n+2} 来说，它的枝链是由 n 个度为 4 的结点，$2n+2$ 个度为 1 的结点和 $3n+1$ 条边构成的树。不同的树的结构对应着不同的化合物。比如上面所示的丁烷与异丁烷就是由于树的结构不同而



(a) (b) (c)

图 2 1-2



(a) CH_4 甲烷　(b) 乙烷　(c) C_3H_8 丙烷

(d) C_4H_{10} 丁烷　(e) C_4H_{10} 异丁烷

图 2-13

各异，这就为化学家们提供了一种很有力的分析物质结构的工具。例如图 2-1-4 所示，两个图从外表看来似乎不尽相同，实际上它们是同构的，因而是同一种物质。

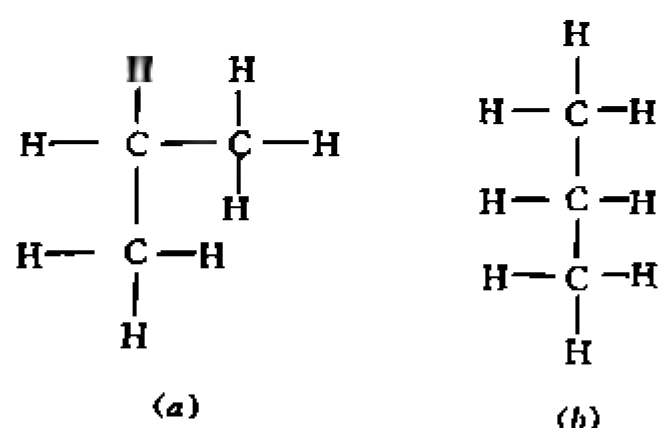


图 2-1-4

例 2：判决树

设有四个银币，已知其中三个是真的，最多有一个是假的。真假的标准在于真的银币重量完全符合标准。现用一天平设法对这四个银币的真假作出判断。

用 a、b、c、d 分别表示四个银币，$a:b$ 表示 a 与 b 在天平上作比较的意思，如图 2-1-5 所示，叶子是判决的结论。

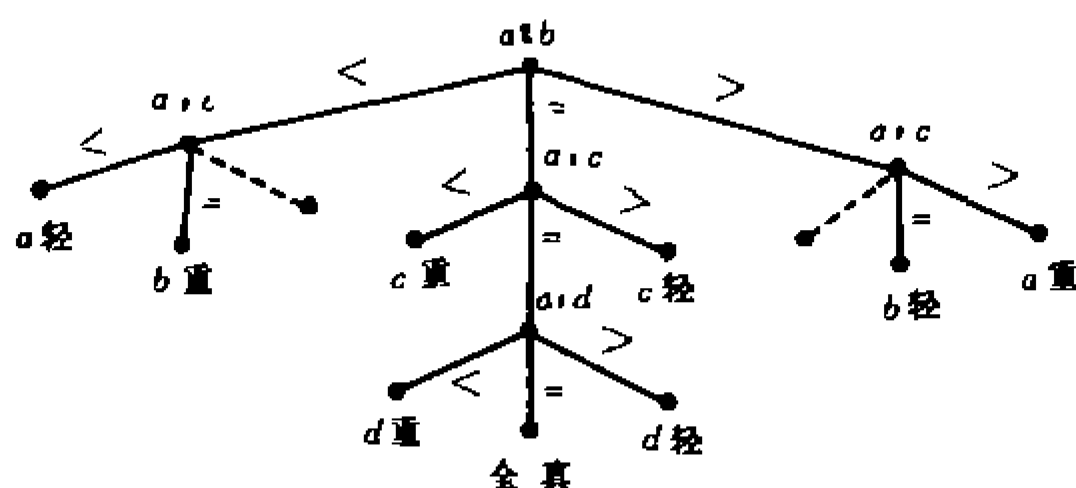


图 2-1-5

上面把作出判决的逻辑关系用树的形式表达出来，使得条理清晰，一目了然。我们称它为判决树。显然办法不是唯一的。比如也可如图 2-1-6 那样的方案进行判决。其中 $a,b:c,d$ 表示 a,b 在天平的一边，c,d 在天平的另一边进行比较。

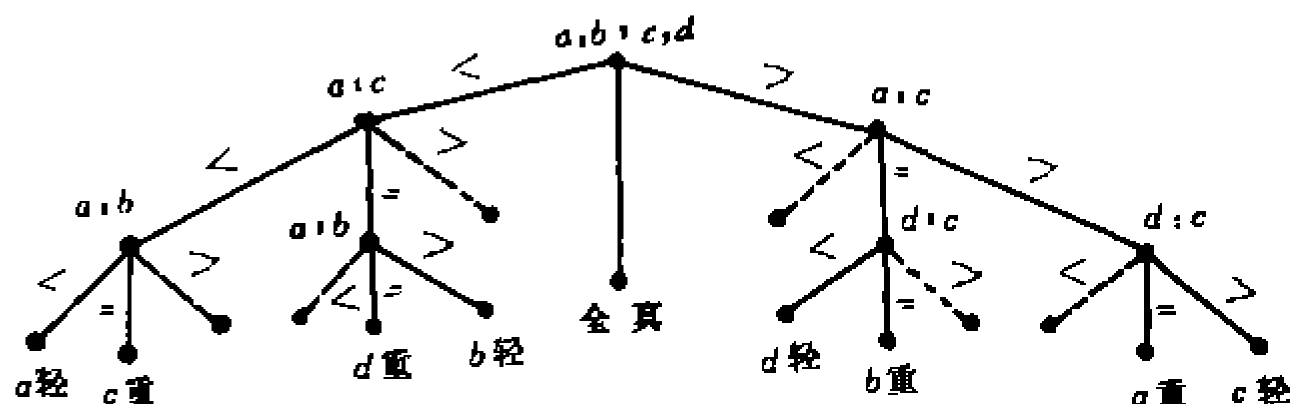


图 2-1-6

若在题中再添一个真币 s，则判决树可取图 2-1-7 形式。

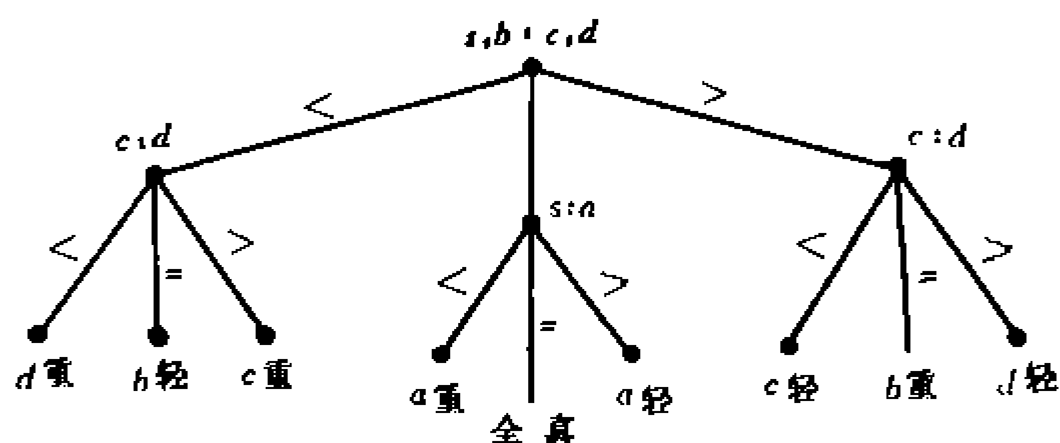


图 2 1-7

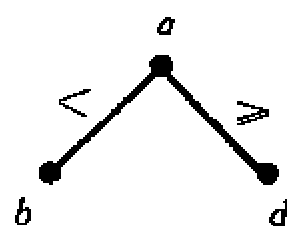


图 2 1-8

例 3：若已给一数列 5，3，7，4，2，9，6，8，1，10。试把它们按次序从小到大排列。

现给出一种算法如图 2-1-8 所表示的那样：即每一顶点有两个分枝，左分枝的数小于 a，而右分枝的数则大于等于 a。按这规律对这数列依次比较得

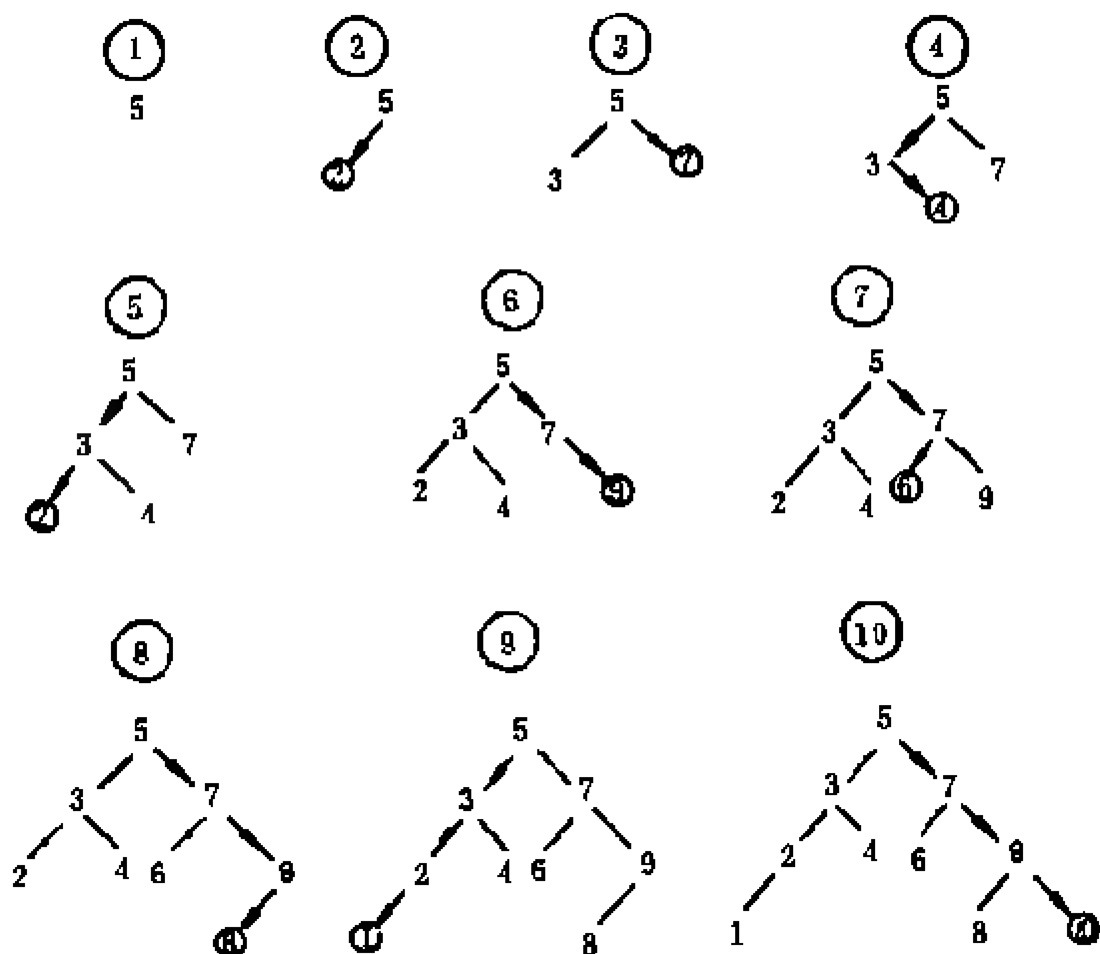


图 2 1 9

最后的数列 1，2，3，4，5，6，7，8，9，10。这种方法在数据结构中广泛应用。

例 4：表达式 $v_1v_2/v_3+v_4(v_5\quad v_6/v_7)$ 可以形象地表示如图 2-1 10。这里(被运算对象)(运算符号)(运算对象)表示以图 2-1-11。

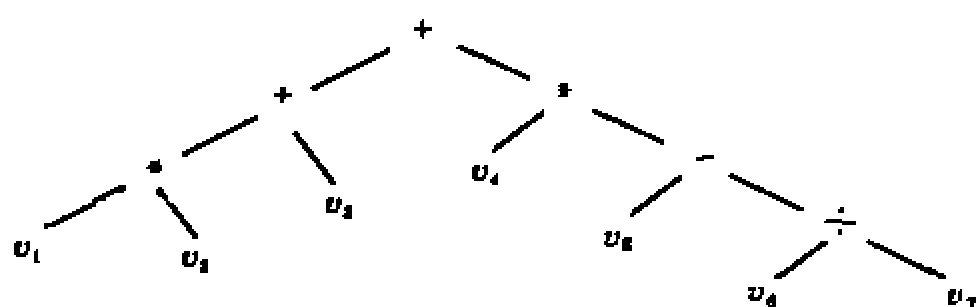


图 2-1-10

图 2-1-11

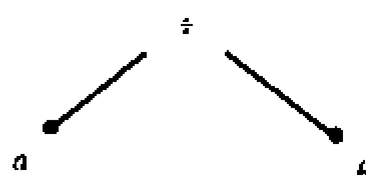


图 2-1-12

从例 4 可知一代数表达式可以用一棵二元树来表达它，称之为该表达式的表达式树。如果图 2-1-11 表示成：

（被运算对象）（运算对象）（运算符号）

则图 2-1-12 写成 $ab\div$。也叫运算符后缀式。根据这个规则，图 2-1-10 可写成：

$$v_1v_2 * v_3 \div v_4v_5v_6v_7 \div - * +$$。

这是波兰表达式。用这种表达式可省去括号。代数表达式的编译过程和把表达式改写成波兰表达式过程相一致。将代数表达改写成波兰表达式实际上是对表达式树按“后缀式”规则的表示。

§2 基本性质

现在我们来看生成树的一些基本性质：

定理 若连通图 $G=(V,E)$，$n=|V|$，则 G 的生成树有 $n-1$ 条边。

证明 用数学归纳法证明如下：

$n=2$ 时，定理显然是对的。设顶点数为 $n-1$ 时，定理为真。若 T 是有 n 个顶点的树，它的所有顶点的度都不为 0，否则将出现孤立点，与连通的假设矛盾。也不可能所有顶点的度都超过 1，否则将出现回路，这与树的定义相矛盾。这就证明了存在一点 v_j，使得：

$$d(v_j)=1$$

从 T 中去掉 v_j 及与 v_j 关联的一条边，余下的图依然是树，而且顶点数为 $n-1$，根据假设，必有 $n-2$ 条边。再把去掉的 v_j 点及其关联边加上去，这就证明了 n 个顶点的树必有 $n-1$ 条边。

对于图 $G=(V,E)$，$|V|=n$，$|E|=m$，下面五个命题是相互等价：

(1) G 是一棵树；

(2) G 的任意两顶点间有且仅有一条道路；

(3) G 是连通的且 $m=n-1$；

(4) G 不包含回路且 $m=n-1$；

(5) G 不含回路而且在不相邻接的两顶点间增加一条边则有且仅有一条回路。

等价性的证明留给读者思考。

§3　关联矩阵与基本关联矩阵

定义　$G=(V,E)$是有向图，其中

$$V=\{v_1,v_2,\cdots,v_n\},\quad E=\{e_1,e_2,\cdots,e_m\}.$$

令：

$$\boldsymbol{B}=(b_{ij})_{n\times m}$$

其中

$$b_{ij}=\begin{cases}1 & (v_i,\ v_k)=e_j\in E;\\ -1 & (v_k,\ v_i)=e_j\in E;\\ 0 & \text{其它。}\end{cases}$$

则称矩阵 B 为有向图 G 的关联矩阵。

例：

$$\boldsymbol{B}=\begin{array}{c}v_1\\v_2\\v_3\\v_4\\ \\ \end{array}\begin{array}{c}\left(\begin{array}{cccccc}1&1&1&0&0&0\\-1&0&0&-1&0&1\\0&0&-1&0&-1&-1\\0&-1&0&1&1&0\end{array}\right)\\ \begin{array}{cccccc}e_1&e_2&e_3&e_4&e_5&e_6\end{array}\end{array}$$

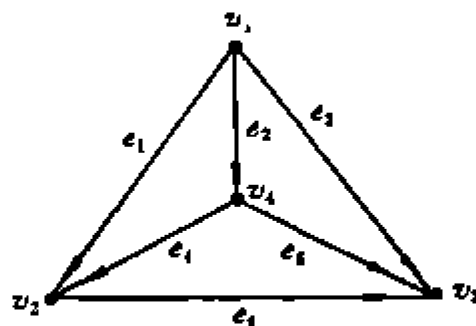


图　2-3-1

定理　连通有向图 $G=(V,E)$，设 $n=|V|$，$m=|E|$，$\boldsymbol{B}$ 是图 G 的关联矩阵，则 $\boldsymbol{B}$ 的秩 rank $\boldsymbol{B}$ 小于 n。

证明　由关联矩阵的定义知矩阵 $\boldsymbol{B}$ 的每一列都有两个非零元素：1 和 -1 组成，故 $\boldsymbol{B}$ 的 n 行的和为零，即 $\boldsymbol{B}$ 的 n 个行向量线性相关，rank $\boldsymbol{B}<n$。定理证毕。

定理　从关联矩阵 $\boldsymbol{B}$ 中任取一个小行列式，它的值不外乎 0，+1，-1。

证明　若 $\boldsymbol{B}_0$ 是 $\boldsymbol{B}$ 的某小方阵，而每列的元素中均有两个非零元素，则 $\det\boldsymbol{B}_0=0$。($\det\boldsymbol{B}_0$ 表示方阵 $\boldsymbol{B}_0$ 的行列式。)

若 $\boldsymbol{B}_0$ 中含有零元素组成的行或列，则 $\det\boldsymbol{B}_0=0$。

若 $\boldsymbol{B}_0$ 中某一行或列只含有一个非零元素。设为 $b_{ij}^{(0)}$，去掉这一元素所在的行和列，余下的元素组成一矩阵 $\boldsymbol{B}_1$，则显然有，

$$\det\boldsymbol{B}_0=\pm\det\boldsymbol{B}_1$$

按这样办法继续进行，每次使矩阵的阶降低一次，直到最后元素为±1 或 0 为止。故 $\det\boldsymbol{B}_0$ 的值不外乎±1 或零，定理得证。

定理　连通有向图 G 的关联矩阵 $\boldsymbol{B}$ 的秩等于 $n-1$。

证明　矩阵 $\boldsymbol{B}$ 的特点是每列有两个非零元素且仅有两个非零元素，一是 +1，一是 -1，其余均为零。利用消元法来判断这 n 行中线性无关的行向量有几？根据 $\boldsymbol{B}$ 矩阵的特

点为消去某一非零元素，只要使该元素所在的列中另一个非零元素所在的行加到该元素所在的行即可。为此先证对于 n 维布尔向量 X 的方程组

$$XB = O$$

的解只有两种可能，一是所有元素均为 1，一是所有元素均为零。显然这两个向量是解。如若不然，设向量 X 的各个分量既不全为 1，也不全为 0。不失一般性，设

$$X = (\underbrace{1\ 1 \cdots 1}_{q}\quad \underbrace{0\ 0 \cdots 0}_{n-q}) = (I_{(q)} \vdots O_{(n-q)})$$

这里 $I_{(q)}$ 表 q 维分量均为 1 的行向量，$O_{(n-q)}$ 表示 $n-q$ 维分量均为 0 的行向量，相应地将 B 分成：

$$B = \underbrace{\begin{pmatrix} B_1 \\ \cdots\cdots \\ B_2 \end{pmatrix}}_{m} \begin{matrix} \}q \\ \\ \}n-q \end{matrix}$$

由于 $XB=0$，即

$$(I_{(q)} \vdots O_{n-q}) \begin{pmatrix} B_1 \\ \cdots\cdots \\ B_2 \end{pmatrix} = 0$$

故

$$I_q B_1 = O$$

即矩阵 B_1 的各列或含有两个非零元素或全为零。不妨设

$$B_1 = (\underbrace{B_{11}}_{p} \vdots \underbrace{O}_{m-p})\}q$$

故

$$B = \underbrace{\begin{pmatrix} B_{11} & O \\ O & B_{22} \end{pmatrix}}_{p\quad m-p} \begin{matrix} \}q \\ \}n-q \end{matrix}$$

当 $0<p<n$ 时，说明若有解 $X=(I_{(q)} \vdots O_{(n-q)})$，则导致图 G 是非连通的，这与假设矛盾。这也就是证明了 B 矩阵的任意 $q(<n)$ 行的和不为零。

若 $p=0$，则

$$B = \begin{pmatrix} O \\ B_{22} \end{pmatrix},$$

若 $p=n$，则

$$B = \begin{pmatrix} B_{11} \\ O \end{pmatrix}。$$

这些都违反了图 G 是连通的假设。这就说明关联矩阵 B 的 n 行和必为零，但是少于 n 行的和必不全为零，否则与连通的假设矛盾。

既然方程 $XB=O$ 的解，或为 $X=I_{(n)}$ 或为 $X=O_{(n)}$，已知 B 的秩小于 n，故 B 的秩为 $n-1$。定理证毕。

定义 从连通图 G 的关联矩阵 B 中，除掉与顶点 v_k 对应的一行，得 $(n-1)\times m$ 的

矩阵 B_k，称为对应于顶点 v_k 的基本关联矩阵。

基本关联矩阵 B_k 是 $(n-1)\times m$ 矩阵，它的秩为 $n-1$，故它的 $n-1$ 行是线性无关的。它的 m 个列向量中也有 $n-1$ 个列是线性无关的。究竟哪些列是独立的列向量呢？下面的定理回答这个问题：

定理 若 B_k 是连通图 G 的基本关联矩阵，而且 $C=\{e_1,e_2,\cdots,e_l\}$ 是某一回路，则回路 C 的各边 $e_1,e_2,\cdots,e_l$ 所对应的矩阵 B_k 中的各列的列向量必线性相关。

证明 回路 C 的长度为 l，矩阵 B_k 中由这 l 条边所对应的 l 个列向量，组成一矩阵 $D=(d_{ij})_{(n-1)\times l}$。若回路 C 包含了 B_k 所对应的顶点 v_k，则 D 有 $l-1$ 行的非零行向量，故 D 的秩 $\leqslant l-1$。若回路不含顶点 v_k，则这 l 列中都含有两个非零元素，故 l 行的和为零。所以 D 的秩依然是 $\leqslant l-1$。

这就证明了这 l 列列向量是线性相关的。

定理 基本关联矩阵 B_k 的任一 $n-1$ 阶行列式不为零的充分必要条件是它的各列对应的边构成图 G 的树。

证明 从上面我们已证 B_k 的秩为 $n-1$，故必有一个 $n-1$ 阶行列式不为零，而且这 $n-1$ 条边必不组成回路。这样有 n 个顶点、$n-1$ 条边，没有回路的图是树，必要性得到证明。

反之，可证明树的各边对应于 B_k 中各列组成的 $n-1$ 阶行列式非零。这只要从 B_k 对应的顶点 v_k 作为始点 v_{k_1}。对顶点和边进行重新编号，保持下面的关系成立：

$$e_j=(v_{j-l_j},v_j)\qquad j=1,2,\cdots,n-1,\ l_j>0。$$

经过重新编号后得矩阵。

$$B_{k_0}=\begin{matrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1}\end{matrix}\begin{bmatrix} \pm 1 & & \boldsymbol{U} & \\ & \pm 1 & & \\ & & & \\ \boldsymbol{0} & & & \pm 1 \end{bmatrix}$$
$$\qquad\qquad e_1\quad e_2\quad \cdots\quad e_{n-1}$$

这矩阵主对角线下面的元素为 0，其行列式不为零，顶点和边的重新编号，只不过对关联矩阵作行列变换，而行列的变换不影响矩阵的秩。定理证毕。

这个定理说明基本关联矩阵 B_k 的 $n-1$ 阶行列式取值 ± 1，当且仅当它的 $n-1$ 列对应一棵树的边，否则为零。

§4 回路矩阵与基本回路矩阵

设连通图 $G=(V,E)$ 的边数为 m，顶点个数为 n，则树 T 必有 $n-1$ 条边；对应的余树的边数为 $m-n+1$。对于树 T 每加上一条属于余树的边，它必然与属于 T 的边形成一条且仅有一条回路，从而可见图 G 至少有 $m-n+1$ 条回路。

如图 2-4-1 为例，边 e_1,e_4,e_5 构成树 T，e_2,e_3,e_6 是对应的余树边，它分别对应回路 $C_1=(e_1,e_2,e_4)$，$C_2=(e_1,e_3,e_5)$，$C_3=(e_4,e_5,e_6)$。

显然图 2-4-1 所示的图 G 的回路不仅这三个，还有如图 2-4-2 所示的回路：

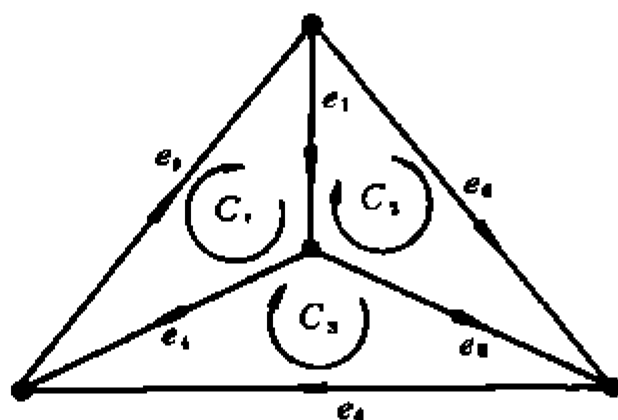


图 2-4-1

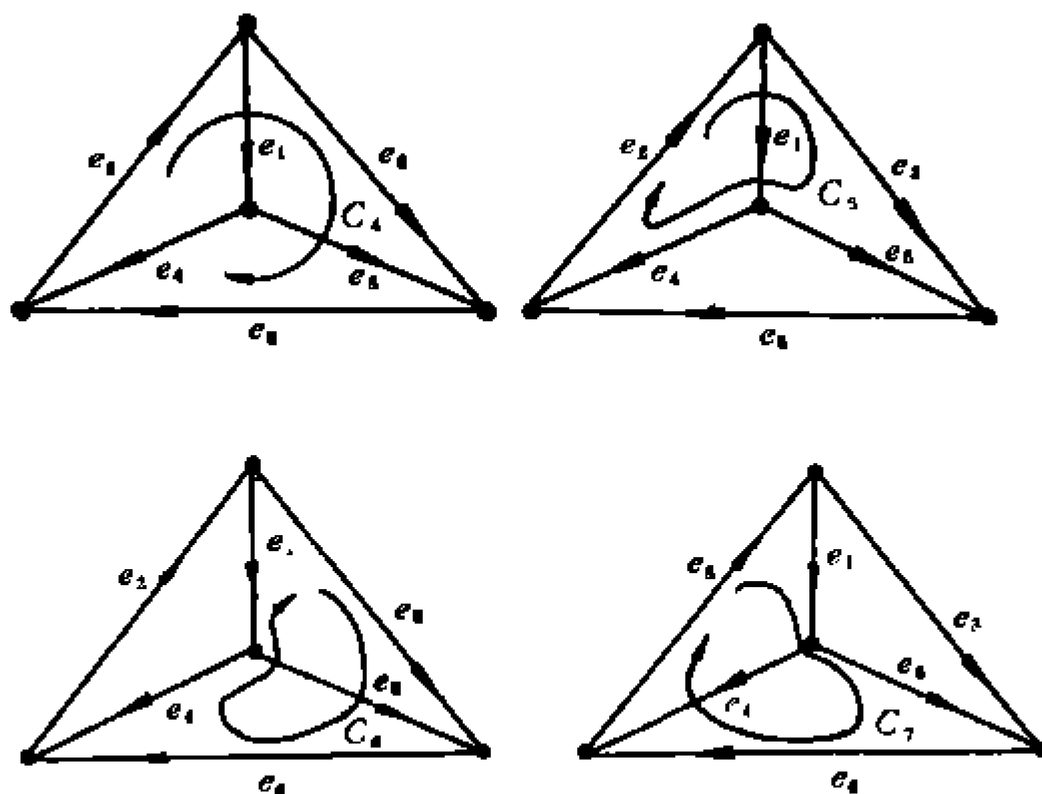


图 2-4-2

引进一矩阵 $C=(c_{ij})_{7\times 6}$，其中

$$c_{ij}=\begin{cases} 1, & \text{若回路 } c_i \text{ 中含有边 } e_j \text{ 且方向一致；} \\ 1, & \text{若回路 } c_i \text{ 中含有边 } e_j \text{ 但方向相反；} \\ 0, & \text{其它。} \end{cases}$$

于是有

$$C=\begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ \end{matrix}\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & -1 & 0 \\ -1 & 0 & 1 & -1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$\quad e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6$$

矩阵 C 叫做回路矩阵。它的行向量显然不是线性独立的。例如，

$C_1=(\ 1\ \ 1\ \ 0\ \ \ 1\ \ \ 0\ \ 0)$； $C_5=(\ 0\ \ 1\ \ 1\ \ \ 1\ \ -1\ \ 0)$；

$C_2=(\ 1\ \ 0\ \ 1\ \ \ 0\ \ -1\ \ 0)$； $C_6=(-1\ \ 0\ \ 1\ \ -1\ \ \ 0\ \ 1)$；

$C_3=(\ 0\ \ 0\ \ 0\ \ -1\ \ \ 1\ \ 1)$； $C_7=(\ 1\ \ 1\ \ 0\ \ \ 0\ \ \ 1\ \ 1)$。

$C_4=(\ 0\ \ 1\ \ 1\ \ \ 0\ \ \ 0\ \ 1)$；

$$C_1+C_2=(1\ 1\ 0\ 1\ 0\ 0)+(-1\ 0\ 1\ 0\ \ 1\ 0)$$
$$=(0\ 1\ 1\ 1\ \ -1\ 0)=C_5$$

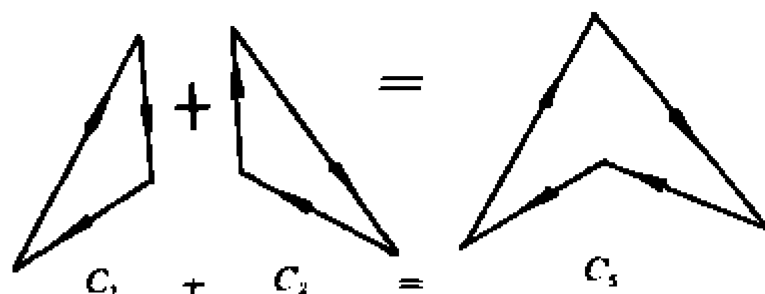


图 2-4-3

类似有

$$C_3+C_5=(0\ 0\ 0-1\ 1\ 1)+(0\ 1\ 1\ 1\ \ \ 1\ 0)$$
$$=(0\ 1\ 1\ 0\ 0\ 1)=C_4。$$

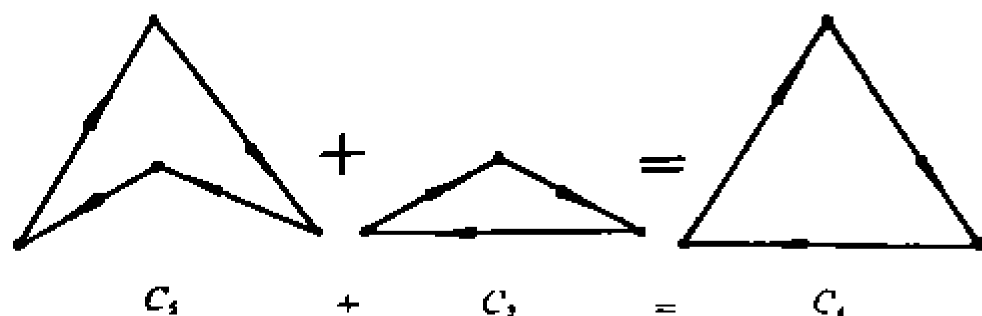


图 2 4 4

直观上不难发现由余树的边对应的回路 C_1,C_2,C_3 三个回路是彼此独立的，而其它的回路可以用它的线性结合来表达，即 C 矩阵的秩是 3。

余树的边所对应的回路叫做基本回路，用 C_f 表示它。例如

$$C_f=\begin{matrix}C_1\\C_2\\C_3\end{matrix}\begin{pmatrix}1&1&0&1&0&0\\1&0&1&0&1&0\\0&0&0&-1&1&1\end{pmatrix}$$
$$\qquad\qquad e_1\ \ e_2\ \ e_3\ \ \ e_4\ \ \ e_5\ \ e_6$$

如若对边的次序重新排列，使余树的边在前面，而树枝在后，而且余数的边依然对应于 C_1,C_2,C_3。故有

$$C_f-\begin{matrix}C_1\\C_2\\C_3\end{matrix}\begin{pmatrix}1&0&0&1&1&0\\0&1&1&-1&0&1\\0&0&1&0&-1&1\end{pmatrix}$$
$$\qquad\qquad e_2\ \ e_3\ \ e_6\ \ \ e_1\ \ \ e_4\ \ e_5$$

这个矩阵的特点是余树边 e_2, e_3, e_6 对应的 3 阶方阵是单位阵。

以上的讨论可以平行地推广到一般平面图的场合。

§5　关联矩阵与回路矩阵的关系

定理　当关联矩阵 $\boldsymbol{B}$ 与回路矩阵 $\boldsymbol{C}$ 中，边的排列次序一致时，恒有：

$$\boldsymbol{BC}^T = \boldsymbol{O}, \qquad \boldsymbol{CB}^T = \boldsymbol{O}。$$

在证明这定理之前先举一个简单的例子看一看究竟是怎么回事，再来证明一般性就不困难了。以图 2-5-1 为例：

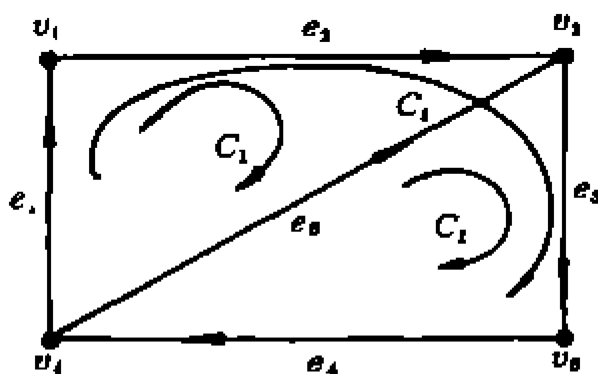


图　2-5 1

$$\boldsymbol{B}=\begin{matrix} v_1 \\ v_2 \\ v_2 \\ v_4 \end{matrix}\begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdot 1 \\ 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & -1 & 1 \end{pmatrix}, \qquad \boldsymbol{C}=\begin{matrix} c_1 \\ c_2 \\ c_3 \end{matrix}\begin{pmatrix} 1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix},$$

$$\qquad e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \qquad\qquad\qquad e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5$$

$$\boldsymbol{BC}^T=\begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & -1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ -1 & 1 & 0 \end{pmatrix}=\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

比如矩阵 $\boldsymbol{B}$ 的第一行与 $\boldsymbol{C}^T$ 的第一列（实际为 C 的第一行）对应元素相乘时，发现 e_1, e_2 与顶点 v_1 相关联，而且同在 C_1 回路上。但对顶点 v_1 来说，e_1 以 v_1 为终点，e_2 以 v_1 为起点，但 e_1 与 e_2 与 C_1 的方向一致，故对应的元素乘积的和为 0。又如 $\boldsymbol{B}$ 的第 一行与 $\boldsymbol{C}$ 的第二行元素相乘时，由于与 v_1 关联的边 e_1, e_2 不在 C_2 上，故结果为零。总之当顶点 v_i 在 C_j 回路上时，与 v_i 关联边必有两条边都在 C_j 上，再者这两条边与 C_j 的方向一致（或相反），则对于顶点 v_i 来说必有一进一出。明白了这个道理，定理的证明就非常清楚了。

定理　连通图 G 的回路矩阵 $\boldsymbol{C}$ 的秩为 $m-n+1$。

证明　前面已证过基本回路矩阵的秩为 $m-n+1$，这实际上已证明了回路矩阵的秩至少为 $m-n+1$。现在只要证明矩阵 $\boldsymbol{C}$ 的秩不超过 $m-n+1$ 就可以了。

上面已证明了 $\boldsymbol{BC}^T=0$，换句话说，$\boldsymbol{C}$ 矩阵的各行都满足方程组：

$$BX = O$$

其中 X 是 m 维列向量。这个方程组有 m 个变元,n 个方程。由于矩阵 B 的秩为 $n-1$,故实际上是 $n-1$ 个独立的 m 元一次方程组。根据线代数理论,这方程组仅有 $m-(n-1)=m-n+1$ 个独立解。也就是说矩阵 C 的各行不会有超过 $m-n+1$ 个线性无关的行向量。定理证毕。

定理 从基本回路矩阵 C_f 中任取 $m-n+1$ 列组成一 $m-n+1$ 阶行列式,它的值非零的充分必要条件是:这些列正好对应于某一余树的 $m-n+1$ 条边。

证明 对于矩阵 C_f 前面已证了余树的边对应的各列组成的行列式不为零,故充分条件已证了。

现在证必要条件,即若从 C_f 中选 $m-n+1$ 列组成的 $m-n+1$ 阶矩阵 C_{11},它的行列式不为零,则这 $m-n+1$ 列对应的边必为一余树。

对边的次序适当排列使得:

$$C_f = (C_{11},\ C_{12}),$$

其中 C_{12} 为 $n-1$ 列。现在只要证这 $n-1$ 列所对应的边不形成回路。因根据树的性质,n 个顶点,$n-1$ 条边,而且没有出现回路的图就是树,这样就证明了 C_{11} 对应的边是余树。

用反证法。若 C_{12} 所对应的各列不是树,即存在这样的一些回路,它仅由 C_{12} 对应的各边构成,则

$$C = \begin{pmatrix} C_f \\ C_k \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ O & C_{22} \end{pmatrix}.$$

这说明 C_{22} 有非零元素,C 的秩大于 $m-n+1$,这与已经证明了的回路矩阵的秩为 $m-n+1$ 相矛盾。证毕。

给出了图 G 的关联矩阵,与无向图的邻接矩阵一样,等于给出了图 G 的全部信息。既然能从图 G 中找出回路也就能从关联矩阵 B 中算出基本回路矩阵来。下面也就讨论基本关联矩阵 B_k 与基本回路矩阵 C_f 之间的关系。

定理 如果图 G 的边是按这样的次序排列,使得:

$$C_f = (\underbrace{I}_{m-n+1}\ \vdots\ \underbrace{C_{12}}_{n-1})\ \}m-n+1,$$

$$B_k = (\underbrace{B_{11}}_{m-n+1}\ \vdots\ \underbrace{B_{12}}_{n-1})\ \}n-1,$$

其中 C_{12} 是 $(m-n+1)\times(n-1)$ 矩阵,B_{12} 是 $(n-1)\times(n-1)$ 矩阵,I 是 $m-n+1$ 阶的单位矩阵。则

$$C_{12} = -B_{11}^T(B_{12}^{-1})^T,\ 或\ C_{12} = -B_{11}^T(B_{12}^T)^{-1},$$

即

$$C_f = (I\ \vdots\ -B_{11}^T(B_{12}^T)^{-1})。$$

证明 由于 $BC^T=0$

$\therefore\quad B_kC_f^T=0$

即

$$(B_{11}\ B_{12})\begin{pmatrix} I \\ C_{12}^T \end{pmatrix} = B_{11} + B_{12}C_{12}^T = 0,$$

$\therefore \quad C_{12}^T = -B_{12}^{-1}B_{11}$,

$C_{12} = -(B_{12}^{-1}B_{11})^T = -B_{11}^T(B_{12}^{-1})^T$。

例： 如图 2-5-2 所示

$$B_4 = \begin{matrix} v_1 \\ v_2 \\ v_3 \end{matrix}\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 & 1 & 0 \end{pmatrix}$$
$$\qquad\quad \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{matrix}$$

$$= \begin{matrix} v_1 \\ v_2 \\ v_3 \end{matrix}\left(\begin{array}{ccc:ccc} 1 & 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 1 & 0 \end{array}\right),$$
$$\qquad\quad \begin{matrix} e_1 & e_6 & e_4 & e_2 & e_5 & e_3 \end{matrix}$$

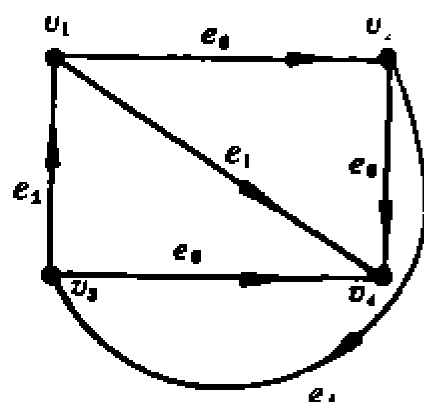


图 2-5-2

$$B_{11} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{pmatrix}, \qquad B_{12} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}。$$

$$B_{12}^{-1} = \begin{pmatrix} -1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}。$$

$$C_{12} = -B_{11}^T(B_{12}^{-1})^T$$
$$= -\begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & -1 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{pmatrix}。$$

$$\therefore C_f = \begin{matrix} C_1 \\ C_2 \\ C_3 \end{matrix}\left(\begin{array}{ccc:ccc} 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & 0 & 1 & -1 \end{array}\right)。$$
$$\qquad\qquad \begin{matrix} e_1 & e_6 & e_4 & e_2 & e_6 & e_3 \end{matrix}$$

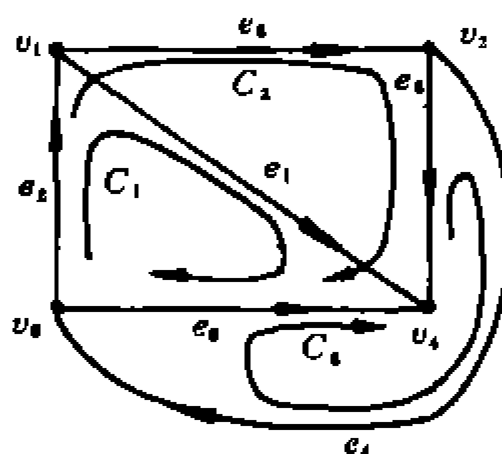


图 2-5-3

对应的基本回路 C_1, C_2, C_3 如图 2-5-3 所示。

§6 割集矩阵与基本割集矩阵

设 $G=(V,E)$ 是连通图，$S\subseteq E$，若从图 G 中消去属于 S 所有的边，则图 $G\backslash S$ 非连通，但消去属于 S 的任何真子集的边，结果保持连通，则称 S 是图 G 的一个割集。也就是说割集 S 是使连通图 G 失去连通性的最小的边集合。

定义 有向图 $G=(V,E)$，$U\subseteq V$，

$$\overrightarrow{E}(U) \triangleq \{e_k \ e_k = (v_i, v_j), v_i \in U, v_j \notin U\}。$$

即 $\overrightarrow{E}(U)$ 为以 U 的元素为始点，终点不属于 U 的边的集合。类似地定义：

$$\overleftarrow{E}(U) \triangleq \{e_k | e_k = (v_i, v_j), v_i \notin U, v_j \in U\}。$$

例：令 $U=\{v_1, v_2\}$，

则

$$\overrightarrow{E}(U) = \{e_2, e_3, e_6\},$$

$$\overleftarrow{E}(U) = \{e_5\}。$$

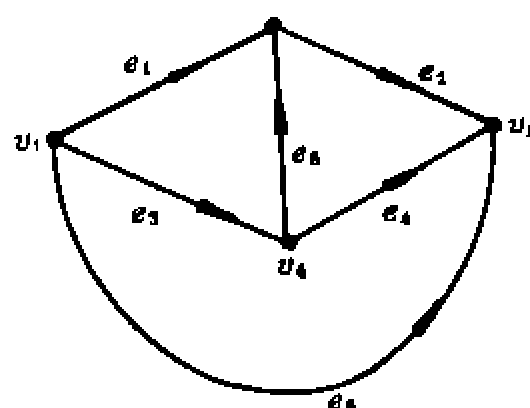


图 2-6-1

定义 有向图 $G=(V,E)$ 的割集 S_k 把顶点 V 分成 V_1, V_2 不相交的两部分，S_k 中边的集合 F_k 有：

$$F_k = \overrightarrow{E}(V_1) \cup \overleftarrow{E}(V_1) = \overrightarrow{E}(V_2) \cup \overleftarrow{E}(V_2)。$$

即 F_k 中有从 V_1 向 V_2 的边和从 V_2 向 V_1 的边。若给这割 S_k 以方向，就称为有向割集。则割集 S_k 中的边分为与 S_k 同向与反向两部分。

如若由 V_1 和 V_2 导出的 G 的子图 $G_1=(V_1,E_1)$，$G_2=(V_2,E_2)$ 是连通的，则由 V_1 和 V_2 确定的割便是割集。图 G 的支撑树 T 可用来确定割集，因为任意消去一树枝，T 便失去连通。

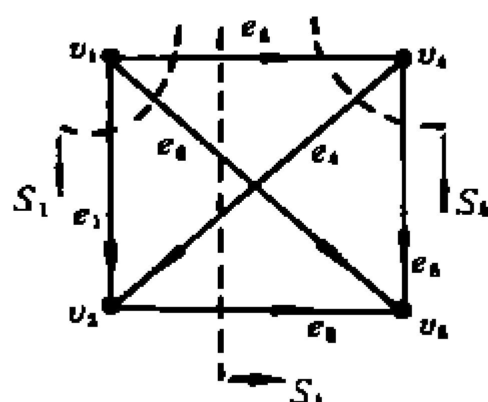


图 2-6-2

定义 T 是有向连通图 G 的树，e_i 是 T 的任一树枝，对应于 e_i 有一有向割集 S_i，S_i 不含有除 e_i 以外别的树枝，而且使得它的方向与 e_i 一致，这样得一组割集 $S_1, S_2, \cdots, S_{n-1}$ 称为基本割集。

如图 2-6-2 虚线所示的割集 S_1, S_2, S_3 分别与树枝 e_1, e_5, e_6 相一致。割集的方向不如回路方向那样容易认识清楚。以 S_2 为例，它包含有边 e_2, e_3, e_4, e_5，把顶点分成 $V_1=\{v_1, v_2\}$，$V_2=\{v_3, v_4\}$ 两部分，S_2 的方向规定为以 V_1 的点为始点，V_2 的点为终点者为正向；反之为负向。故与 S_2 同向的有 e_2, e_3, e_5；与 S_2 反向的有 e_4。

割集矩阵：

设 $S_1, S_2, \cdots, S_k$ 是图 $G=(V,E)$ 的割集，矩阵 $S=(s_{ij})_{k\times m}$，

其中

$$S_{ij} = \begin{cases} 1, & \text{割集 } S_i \text{ 含有边 } e_j \text{ 并同向；} \\ -1, & \text{割集 } S_i \text{ 含有边 } e_j \text{ 并反向；} \\ 0, & \text{其它。} \end{cases}$$

则称 S 为割集矩阵，基本割集 $S_1, S_2, \cdots, S_{n-1}$ 对应的割集矩阵称为基本割集矩阵 S_f。

例：图 2-6-3 所示的割集的割集矩阵为：

$$
S=\begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \end{matrix}\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 1 & -1 & 1 \\ 0 & -1 & 1 & 1 & 0 & -1 \end{pmatrix}。
$$
$$
\quad\quad\quad e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6
$$

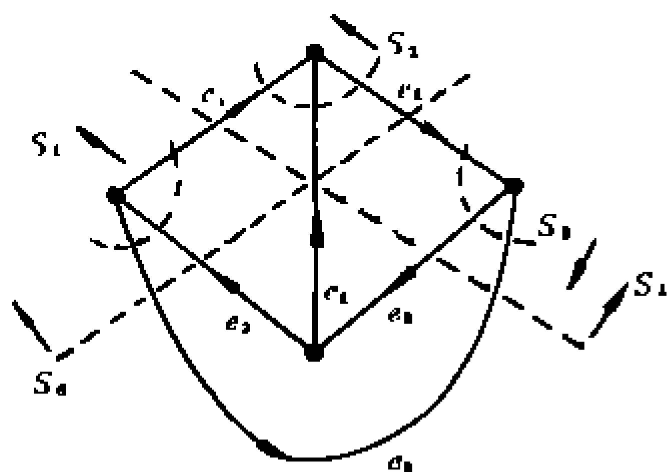


图 2-6-3

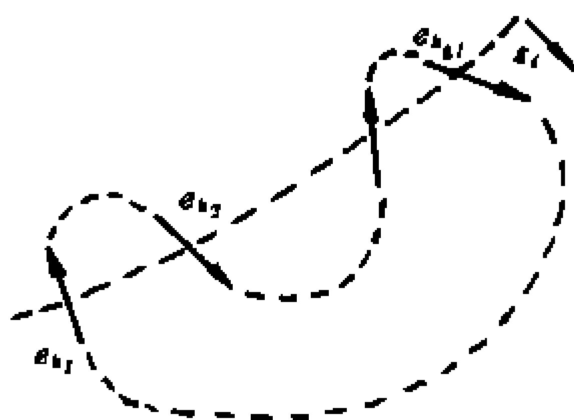


图 2-6-4

对应于树 $T=(e_3,e_4,e_5)$的基本割集矩阵为：

$$
S_f=\begin{matrix} S_1 \\ S_2 \\ S_3 \end{matrix}\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & -1 \end{pmatrix}。
$$
$$
\quad\quad\quad e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6
$$

定理 当割集矩阵 S 与回路矩阵 C 的边的次序一致时，恒有：

$$
CS^T=O, \qquad SC^T=O。
$$

证明 若 G 图的任一割集与某一回路 C 有共同边时，则共同边的数目必为偶数。

设有向割集 S_i 与回路 C_j 有 $2l$ 条共同边，令它沿回路 C 依次为 $e_{k_1},e_{k_2},\cdots,e_{k_{2l}}$。如果边 e_{k_1} 与 e_{k_2} 在回路中方向一致，则它们在 S_i 中的方向必相反，也就是说若 e_{k_1},e_{k_2} 与 C 同向，则其中之一与 S_i 同向，另一个与 S_i 反向。于是求 S 的第 i 行与 C^T 的第 j 列的对应元素乘积的和时，有

$$
\sum_{k=1}^{m} s_{ik}c_{jk}=\sum_{p=1}^{2l} s_{ik_p}c_{jk_p}=0
$$

定理证毕。

定理 矩阵 S 的秩为 $n-1$。

证明 基本割集矩阵的秩为 $n-1$，故割集矩阵 S 的秩至少为 $n-1$。只要证 S 的秩不超过 S_f 就可以了。另一方面 $CS^T=O$，而 C 的秩为 $m-n+1$。从方程组：

$$
CX=O
$$

来看，S^T 的每一列向量都是这方程组的解，根据线性代数理论，这方程组仅有 $m-(m-n+1)$ 个独立解，即只有 $n-1$ 个独立解；这就证明了 S 的秩不超过 $n-1$。定理得证。

将连通图 G 的边的次序适当调整，让余树的边在前，树枝在后，使得基本割集矩阵 S_f

和基本回路矩阵 C_f 分别具有下列形式：

$$S_f = (\underbrace{S_{11}}_{m-n+1} \vdots \underbrace{I_{(n-1)}}_{n-1}) \}n-1,$$

$$C_f = (\underbrace{I_{(m-n+1)}}_{m-n+1} \vdots \underbrace{C_{12}}_{n-1}) \}m-n+1,$$

其中 $I_{(n-1)}$ 为 $n-1$ 阶单位矩阵。则有：

定理 $S_{11} = -C_{12}^T$。

证明 从 $S_f^T C_f = O$ 有

$$(S_{11} \quad I_{(n-1)})\begin{pmatrix} I_{(m-n+1)} \\ C_{12}^T \end{pmatrix} = O,$$

$$\therefore \quad S_{11} + C_{12}^T = O。$$

定理得证。

例：如图 2-6-5 所示：

$$S_f = \begin{matrix} S_1 \\ S_2 \\ S_3 \\ \\ \end{matrix}\begin{pmatrix} 1 & 1 & 0 & \vdots & 1 & 0 & 0 \\ 1 & 1 & -1 & \vdots & 0 & 1 & 0 \\ 0 & 1 & 1 & \vdots & 0 & 0 & 1 \end{pmatrix}$$

$$\qquad\quad e_1 \quad e_6 \quad e_4 \quad e_2 \quad e_5 \quad e_3$$

$$= (S_{11}\ I)$$

$$\therefore \quad -C_{12}^T = S_{11} = \begin{pmatrix} -1 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix},$$

$$\therefore \quad C_{12} = \begin{pmatrix} 1 & -1 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix}。$$

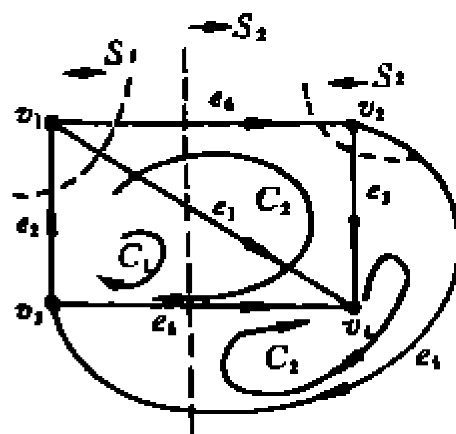


图 2-6-5

$$C = (I \vdots C_{12})$$

$$= \begin{pmatrix} 1 & 0 & 0 & \vdots & 1 & 1 & 0 \\ 0 & 1 & 0 & \vdots & 1 & -1 & 1 \\ 0 & 0 & 1 & \vdots & 0 & 1 & -1 \end{pmatrix}\begin{matrix} C_1 \\ C_2 \\ C_3 \end{matrix}。$$

$$\quad e_1 \quad e_6 \quad e_4 \quad e_2 \quad e_5 \quad e_3$$

回路 C_1、C_2、C_3 如图 2-6-5 所示。

§7 树的数目

首先介绍一个在下面研究中扮演着重要角色的 Binet-Cauchy 定理。

定理：已知 $A=(a_{ij})_{m\times n}$，$B=(b_{ij})_{n\times m}$，且 $m\leqslant n$，则

$$\det(AB) = \sum_{(j)} A_j B_j。$$

其中 A_j、B_j 是 m 阶行列式，A_j 是从矩阵 A 中取第 $j_1, j_2, \cdots, j_m$ 列组成的行列式；而 B_j 正好是从 B 中取相应的第 $j_1, j_2, \cdots, j_m$ 行组成的行列式；$\sum_{(j)}$ 是对所有排列 $j_1, j_2, \cdots, j_m$ 求和。

即

$$\det(\boldsymbol{AB})=\sum_{1\leqslant j_1<j_2<\cdots<j_m\leqslant n}\begin{vmatrix}a_{1j_1} & a_{1j_2} & \cdots & a_{1j_m}\\ a_{2j_1} & a_{2j_2} & \cdots & a_{2j_m}\\ \cdots & \cdots & \cdots & \cdots\\ a_{mj_1} & a_{mj_2} & \cdots & a_{mj_m}\end{vmatrix}\begin{vmatrix}b_{j_11} & b_{j_12} & \cdots & b_{j_1m}\\ b_{j_21} & b_{j_22} & \cdots & b_{j_2m}\\ \cdots & \cdots & \cdots & \cdots\\ b_{j_m1} & b_{j_m2} & \cdots & b_{j_mm}\end{vmatrix}。$$

在证明这定理之前先举个例子说明如下：

$$\boldsymbol{A}=\begin{pmatrix}1 & 2 & 1\\ -1 & 3 & 1\end{pmatrix},\qquad \boldsymbol{B}=\begin{pmatrix}2 & 1\\ 0 & 1\\ 1 & 1\end{pmatrix}。$$

$$\boldsymbol{AB}=\begin{pmatrix}1 & 2 & 1\\ -1 & 3 & 1\end{pmatrix}\begin{pmatrix}2 & 1\\ 0 & 1\\ 1 & 1\end{pmatrix}=\begin{pmatrix}3 & 4\\ -1 & 3\end{pmatrix}。$$

$$\det(\boldsymbol{AB})=\begin{vmatrix}3 & 4\\ -1 & 3\end{vmatrix}=13$$

根据 Binet-Cauchy 定理可知：

$$\det(\boldsymbol{AB})=\begin{vmatrix}3 & 4\\ -1 & 3\end{vmatrix}=\begin{vmatrix}1 & 2\\ -1 & 3\end{vmatrix}\begin{vmatrix}2 & 1\\ 0 & 1\end{vmatrix}+\begin{vmatrix}1 & 1\\ -1 & 1\end{vmatrix}\begin{vmatrix}2 & 1\\ 1 & 1\end{vmatrix}+\begin{vmatrix}2 & 1\\ 3 & 1\end{vmatrix}\begin{vmatrix}0 & 1\\ 1 & 1\end{vmatrix}$$

$$=5\cdot 2+2\cdot 1+(-1)\cdot(-1)=13。$$

下面给出的 Binet-Cauchy 定理的证明是初等的，但也比较冗长，故非必要可以考虑省略。在证明之前先叙述行列式的 Laplace 展开法如下。设

$$\boldsymbol{A}=\begin{pmatrix}a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\\ \cdots & \cdots & \cdots & \cdots\\ a_{n1} & a_{n2} & \cdots & a_{nn}\end{pmatrix},$$

$$\boldsymbol{A}\begin{pmatrix}p_1 & p_2 & \cdots & p_l\\ q_1 & q_2 & \cdots & q_l\end{pmatrix}=\begin{vmatrix}a_{p_1q_1} & a_{p_1q_2} & \cdots & a_{p_1q_l}\\ a_{p_2q_1} & a_{p_2q_2} & \cdots & a_{p_2q_l}\\ \cdots & \cdots & \cdots & \cdots\\ a_{p_lq_1} & a_{p_lq_2} & \cdots & a_{p_lq_l}\end{vmatrix},$$

其中 $1\leqslant p_1<p_2<\cdots<p_l\leqslant n$，$1\leqslant q_1<q_2<\cdots<q_l\leqslant n$。从 1 到 n 的正整数中除去 $p_1,p_2,\cdots,p_l$ 后剩下的是 $r_1<r_2<\cdots<r_{n-l}$。同样从 1 到 n 的正整数除去 $q_1,q_2,\cdots,q_l$ 后剩下的是 $s_1<s_2<\cdots<s_{n-l}$。令

$$\boldsymbol{A}'\begin{pmatrix}p_1 & p_2 & \cdots & p_l\\ q_1 & q_2 & \cdots & q_l\end{pmatrix}=(-1)^{\sum_{j=1}^{l}p_j+\sum_{j=1}^{l}q_j}\boldsymbol{A}\begin{pmatrix}r_1 & r_2 & \cdots & r_{n-l}\\ s_1 & s_2 & \cdots & s_{n-l}\end{pmatrix}$$

在确定了 $p_1,p_2,\cdots,p_l$ 后，行列式 $\boldsymbol{A}$ 的 Laplace 展开式为：

$$\boldsymbol{A}=\sum_{q_1<q_2<\cdots<q_l}\boldsymbol{A}\begin{pmatrix}p_1 & p_2 & \cdots & p_l\\ q_1 & q_2 & \cdots & q_l\end{pmatrix}\boldsymbol{A}'\begin{pmatrix}p_1 & p_2 & \cdots & p_l\\ q_1 & q_2 & \cdots & q_l\end{pmatrix}。$$

它的证明见诸高等代数。

例：

$$\begin{vmatrix}1&-9&-2&3\\-5&5&3&-2\\12&-6&1&1\\9&0&-2&1\end{vmatrix}=(-1)^{1+2+1+2}\begin{vmatrix}-1&-9\\-5&5\end{vmatrix}\cdot\begin{vmatrix}1&1\\-2&1\end{vmatrix}$$

$$+(-1)^{1+2+1+3}\begin{vmatrix}-1&-2\\-5&3\end{vmatrix}\cdot\begin{vmatrix}-6&1\\0&1\end{vmatrix}$$

$$+(-1)^{1+2+1+4}\begin{vmatrix}-1&3\\-5&2\end{vmatrix}\cdot\begin{vmatrix}-6&1\\0&-2\end{vmatrix}$$

$$+(-1)^{1+2+2+3}\begin{vmatrix}-9&-2\\5&3\end{vmatrix}\cdot\begin{vmatrix}12&1\\9&1\end{vmatrix}$$

$$+(-1)^{1+2+2+4}\begin{vmatrix}-9&3\\5&-2\end{vmatrix}\cdot\begin{vmatrix}12&1\\9&2\end{vmatrix}$$

$$+(-1)^{1+2+3+4}\begin{vmatrix}-2&3\\3&-2\end{vmatrix}\cdot\begin{vmatrix}12&-6\\9&0\end{vmatrix}$$

$$=(-50)(3)-(-13)\cdot(-6)+(17)\cdot(12)$$

$$+(-17)\cdot(-21)-(3)(15)+(-5)(54)$$

$$=18$$

不难看出通常沿一行展开行列式的方法是 Laplace 展开法的特殊情形。

下面开始转入定理的证明。由于

$$\begin{pmatrix}I_{(m)}&A\\O&I_{(n)}\end{pmatrix}\begin{pmatrix}A&O\\-I_{(n)}&B\end{pmatrix}=\begin{pmatrix}O&AB\\-I_{(n)}&B\end{pmatrix}$$

而且根据 Laplace 展开法可得：

$$\det\begin{pmatrix}I_{(m)}&A\\O&I_{(n)}\end{pmatrix}=1$$

所以

$$\det\begin{pmatrix}A&O\\-I_{(n)}&B\end{pmatrix}=\det\begin{pmatrix}O&AB\\-I_{(n)}&B\end{pmatrix}$$

上式右端根据 Laplace 定理展开，当 $p_1=1, p_2=2, \cdots, p_m=m$ 取定之后，只有 $q_1=n+1, q_2=n+2, \cdots, q_m=n+m$ 时展开项才有贡献。故得

$$\det\begin{pmatrix}A&O\\-I_{(n)}&B\end{pmatrix}=(-1)^{1+2+\cdots+m+(m+1)+(m+2)+\cdots+(m+n)}$$

$$=(-1)^{1+2+\cdots+m+(n+1)+(n+2)+\cdots+(n+m)}(-1)^{n}\det(AB)$$

$$=(-1)^{n+\frac{1}{2}m(m+1)+mn+\frac{m}{2}(m+1)}\det(AB)$$

$$=(-1)^{(n+m)(m+1)}\det(AB)。$$

但

$$
\begin{pmatrix} A & O \\ -I_{(n)} & B \end{pmatrix} = \left(\begin{array}{cccc|cccc}
a_{11} & a_{12} & \cdots & a_{1n} & & & & \\
a_{21} & a_{22} & \cdots & a_{2n} & & & & \\
\cdots & \cdots & \cdots & & & O & & \\
a_{m1} & a_{m2} & \cdots & a_{mn} & & & & \\
\hline
-1 & & & & b_{11} & b_{12} & \cdots & b_{1m} \\
 & -1 & & 0 & b_{21} & b_{22} & \cdots & b_{2m} \\
 & 0 & \ddots & & & \cdots & \cdots & \cdots \\
 & & & -1 & b_{n1} & b_{n2} & \cdots & b_{nm}
\end{array}\right)
\begin{array}{l} \left.\vphantom{\begin{array}{c}a\\a\\a\\a\end{array}}\right\} m \\ \left.\vphantom{\begin{array}{c}a\\a\\a\\a\end{array}}\right\} n \end{array}
$$

$\underbrace{\qquad\qquad}_{n}\quad\underbrace{\qquad\qquad}_{m}$

取 $p_1=1, p_2=2, \cdots, p_m=m$ 对上式的右端的行列式实行 Laplace 展开，并注意若选取第 $j_1, j_2, \cdots, j_m$ 列的项为

$$
(-1)^{\frac{m(m+1)}{2}+\sum\limits_{k=1}^{m} j_k}
\begin{vmatrix}
a_{1j_1} & a_{1j_2} & \cdots\cdots & a_{1j_m} \\
a_{2j_1} & a_{2j_2} & \cdots\cdots & a_{2j_m} \\
\cdots & \cdots & & \\
a_{mj_1} & a_{mj_2} & \cdots\cdots & a_{mj_m}
\end{vmatrix}
\cdot
\left|\begin{array}{c|cccc}
 & b_{11} & b_{12} & \cdots & b_{1m} \\
\underbrace{B_{11}}_{n-m} & b_{21} & b_{22} & \cdots & b_{2m} \\
 & \cdots & \cdots & \cdots & \\
 & b_{n1} & b_{n2} & \cdots & b_{nm}
\end{array}\right| 。 \tag{1}
$$

其中矩阵 B_{11} 是 $n\times(n-m)$ 矩阵，只有 $n-m$ 个 (-1) 元素，其余元素均为零。而这 $n-m$ 个非零元素分布在从上到下，从左到右除去第 $j_1, j_2, \cdots, j_m$ 行以外的 $n-m$ 行里。沿这 $n-m$ 行采用 Laplace 展开式可得：

$$
\left|\begin{array}{c|cccc}
 & b_{11} & b_{12} & \cdots & b_{1m} \\
 & b_{21} & b_{22} & \cdots & b_{2m} \\
\underbrace{B_{11}}_{n-m} & \cdots & \cdots & \cdots & \\
 & b_{n1} & b_{n2} & \cdots & b_{nm}
\end{array}\right|
$$

$$
=(-1)^{n-m+\sum\limits_{k=1}^{m} j_k+(n-m)m+\frac{m(m+1)}{2}}
\begin{vmatrix}
b_{j_1 1} & b_{j_1 2} & \cdots & b_{j_1 m} \\
b_{j_2 1} & b_{j_2 2} & \cdots & b_{j_2 m} \\
\cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\
b_{j_m 1} & b_{j_m 2} & \cdots & b_{j_m m}
\end{vmatrix} 。
$$

故公式(1)变为：

$$
(-1)^{2\sum\limits_{k=1}^{m} j_k+m(m+1)+(n-m)(m+1)} \cdot
\begin{vmatrix}
a_{1j_1} & a_{1j_2} & \cdots & a_{1j_m} \\
a_{2j_1} & a_{2j_2} & \cdots & a_{2j_m} \\
\cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\
a_{mj_1} & a_{mj_2} & \cdots & a_{mj_m}
\end{vmatrix}
\cdot
\begin{vmatrix}
b_{j_1 1} & b_{j_1 2} & \cdots & b_{j_1 m} \\
b_{j_2 1} & b_{j_2 2} & \cdots & b_{j_2 m} \\
\cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\
b_{j_m 1} & b_{j_m 2} & \cdots & b_{j_m m}
\end{vmatrix}
$$

$$
=(-1)^{(n-m)(m+1)}\begin{vmatrix} a_{1j_1} & a_{1j_2} & \cdots & a_{1j_m} \\ a_{2j_1} & a_{2j_2} & \cdots & a_{2j_m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{mj_1} & a_{mj_2} & \cdots & a_{mj_m} \end{vmatrix} \cdot \begin{vmatrix} b_{j_11} & b_{j_12} & \cdots & b_{j_1m} \\ b_{j_21} & b_{j_22} & \cdots & b_{j_2m} \\ \cdots & \cdots & \cdots & \cdots \\ b_{j_m1} & b_{j_m2} & \cdots & b_{j_mm} \end{vmatrix}
$$

$$
\therefore (-1)^{(n+m)(m+1)}\det(AB) = (-1)^{(n-m)(m+1)}\sum_{(i)} A_iB_i
$$

$$
\det(AB) = (-1)^{2(m+1)n}\cdot\sum_{(i)} A_iB_i = \sum_{(i)} A_iB_i
$$

定理 设 B_k 是连通有向图 G 的基本关联矩阵，则图 G 的生成树的数目为：

$$
\det(B_kB_k^T)。
$$

证明 根据 Binet Cauchy 定理知道：当

$$
A = B_k, \qquad B = B_k^T
$$

时有

$$
\det(B_kB_k^T) = \sum_{1\leqslant j_1<\cdots<j_{n-1}\leqslant m}\begin{vmatrix} b_{1j_1} & b_{1j_2} & \cdots & b_{1j_{n-1}} \\ b_{2j_1} & b_{2j_2} & \cdots & b_{2j_{n-1}} \\ b_{(n-1)j_1} & b_{(n-1)j_2} & \cdots & b_{(n-1)j_{n-1}} \end{vmatrix}^2。
$$

而

$$
\begin{vmatrix} b_{1j_1} & b_{1j_2} & \cdots & b_{1j_{n-1}} \\ b_{2j_1} & b_{2j_2} & \cdots & b_{2j_{n-1}} \\ b_{n-1j_1} & b_{n-1j_2} & \cdots & b_{n-1j_{n-1}} \end{vmatrix} = \begin{cases} \pm 1, (e_{j_1}, e_{j_2}\cdots, e_{j_{n-1}}) \text{ 是树；} \\ 0, \quad \text{其它。} \end{cases}
$$

$$
\therefore \quad \det(B_kB_k^T) = \sum (\pm 1)^2 = \text{树的数目。}
$$

例：如图 2-7-1 所示

$$
B = \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \end{matrix}\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 \end{bmatrix}
$$
$$
\qquad\quad e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6
$$

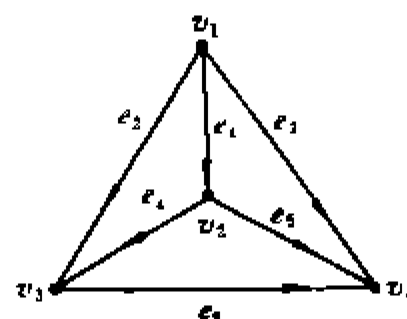


图 2-7-1

任取一基本关联矩阵：

$$
B_4 = \begin{matrix} v_1 \\ v_2 \\ v_3 \end{matrix}\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \end{pmatrix}。
$$
$$
\qquad\quad e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6
$$

$$B_4B_4^T=\begin{pmatrix}1&1&1&0&0&0\\-1&0&0&1&1&0\\0&-1&0&-1&0&1\end{pmatrix}\begin{pmatrix}1&1&0\\1&0&1\\1&0&0\\0&1&-1\\0&1&0\\0&0&1\end{pmatrix}=\begin{pmatrix}3&1&1\\-1&3&-1\\-1&1&3\end{pmatrix}$$

所以
$$\det(B_4B_4^T)=16。$$

现把 16 棵树的图象列表如图 2-7-2：

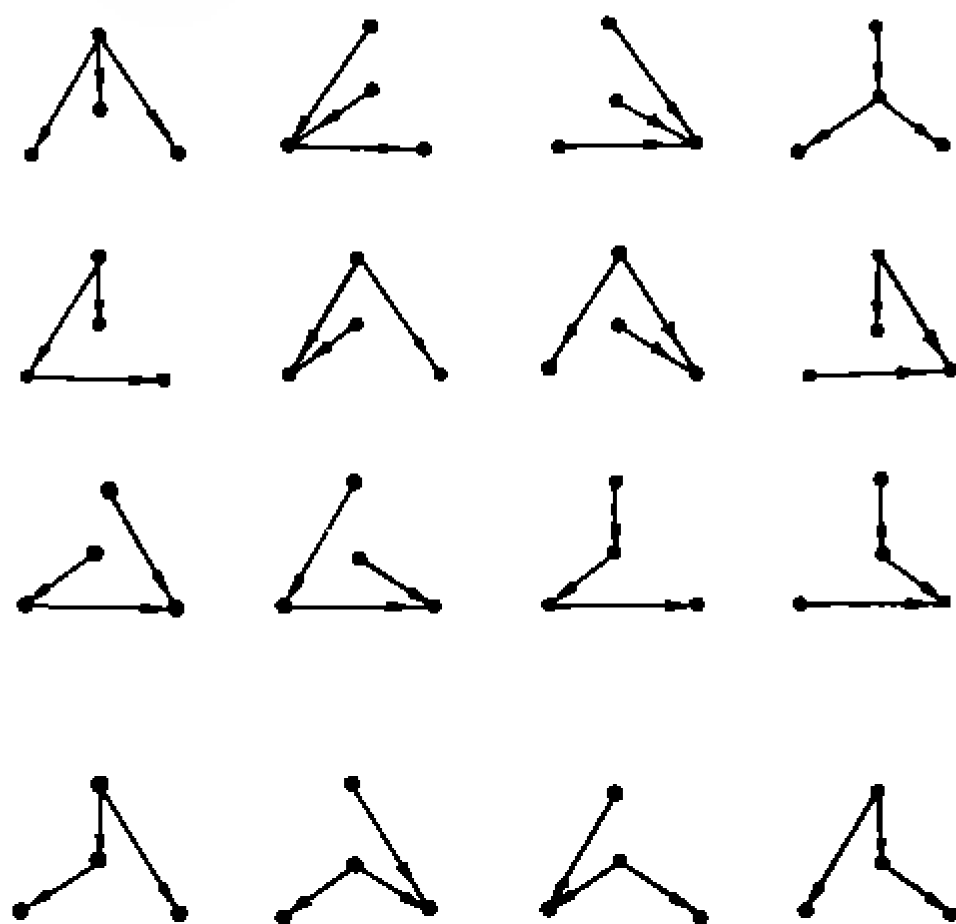


图 2-7-2

§8　内向树与外向树

内向树与外向树是树的两种特殊类型，首先介绍树的概念。为此先引入一个记号：v_i，v_j 是有向图 G 的两个顶点，若存在道路 $(e_1,e_2,\cdots,e_k)$，从 v_i 到达 v_j，我们用符号 $v_i<v_j$ 来表示。

定义　对于有向树 T，若存在一顶点 v_0，使得这树的其它顶点 v_i，恒有 $v_0<v_i$ 时，就称这个树是以 v_0 为始点的外向树，用 $\overrightarrow{T}_0$ 表示它。

定义　对于有向树 T，若存在一顶点 v_0，使得这树的其它顶点 v_i，恒有 $v_i<v_0$，就称这个树是以 v_0 为终点的内向树，用 $\overleftarrow{T}_0$ 表示它。

外向树的特征是：始点除外，每一个顶点都有一条边进入，且仅有一条边进入，即入度为 1。同理内向树的特征是：终点除外，每个顶点都有一条边从该点出去且仅有一条边从

该点出去，即出度为 1。

例：

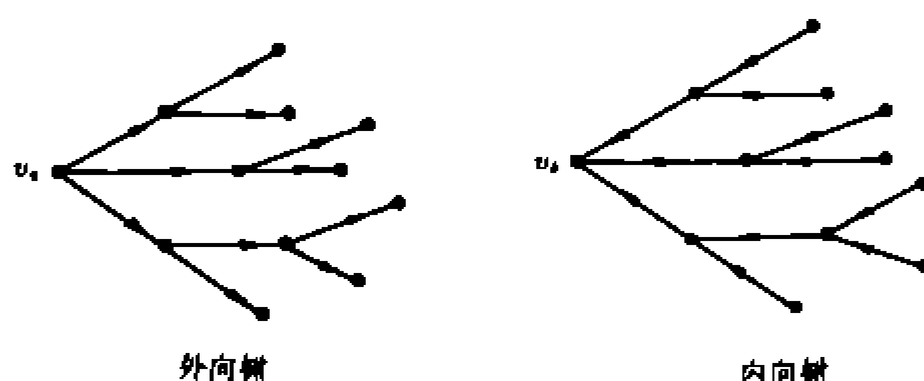


图 2-8-1

从上可知，对于外向树，关于始点 v_a 的基本关联矩阵中，任一外向树“树枝”所对应的 $(n-1)\times(n-1)$ 矩阵中，每一行都有且仅有一个 (-1) 元素，这样 $n-1$ 个 (-1) 元素分布在各列中。换句话说，每行每列都有且仅有一个 (-1) 元素。

类似的道理，对于内向树来说，关于终点 v_b 的基本关联矩阵中，对应于内向树树枝的 $(n-1)\times(n-1)$ 矩阵中，每行每列有且仅有一个 $(+1)$ 元素。

例：

$$B=\begin{array}{c}v_1\\v_2\\v_3\\v_4\\ \\\end{array}\begin{array}{c}\begin{pmatrix}1&1&1&0&0&0\\1&0&0&1&1&0\\0&-1&0&-1&0&1\\0&0&-1&0&-1&-1\end{pmatrix}\\ \begin{matrix}e_1&e_2&e_3&e_4&e_5&e_6\end{matrix}\end{array},$$

$$B_1=\begin{pmatrix}-1&0&0&1&1&0\\0&-1&0&1&0&1\\0&0&-1&0&-1&-1\end{pmatrix}。$$

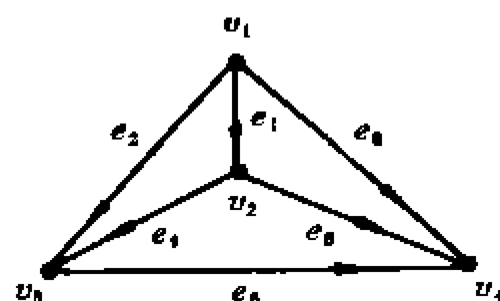


图 2-8-2

以 v_1 为顶点的外向树和它对应的方阵见图 2-8-3。

以外向树为例，若设始点为 v_0，对顶点与边重新编号使得：

$$e_j=(v_{j-l_j},v_j),l_j>0,j=1,2,\cdots,n-1,$$

这样外向树 $\vec{T}_0$ 对应的 $(n-1)\times(n-1)$ 方阵，是上三角矩阵，即主对角线以下的元素均为 0。而且主对角线上的元素为 (-1)。不难发现，若令所有 $(+1)$ 元素改为零，不改变这行列式的值。换句话说外向树 T_0 对应的行列式的值与把 1 改为零后的行列式的值相等。若不存在以 v_0 为顶点的外向树时，其结果如何？留给读者思考。

外向树也是树，所以应该满足树的条件，即基本关联矩阵对应的树枝的 $(n-1)\times(n-1)$ 矩阵，其行列式的值为 ±1。

对 v_k 为始点的外向树，基本关联矩阵 B_k 中，把元素 $(+1)$ 改为零后得一矩阵 $\vec{B}_k$。请注意到这样的事实：从 B_k 中选 $n-1$ 列组成一 $n-1$ 阶行列式，这行列式的值当而且仅当这 $n-1$ 列所对应的边是一棵树时才不为零，而为 ±1。当 $\vec{B}_k$ 中相应的行列式不为零时，

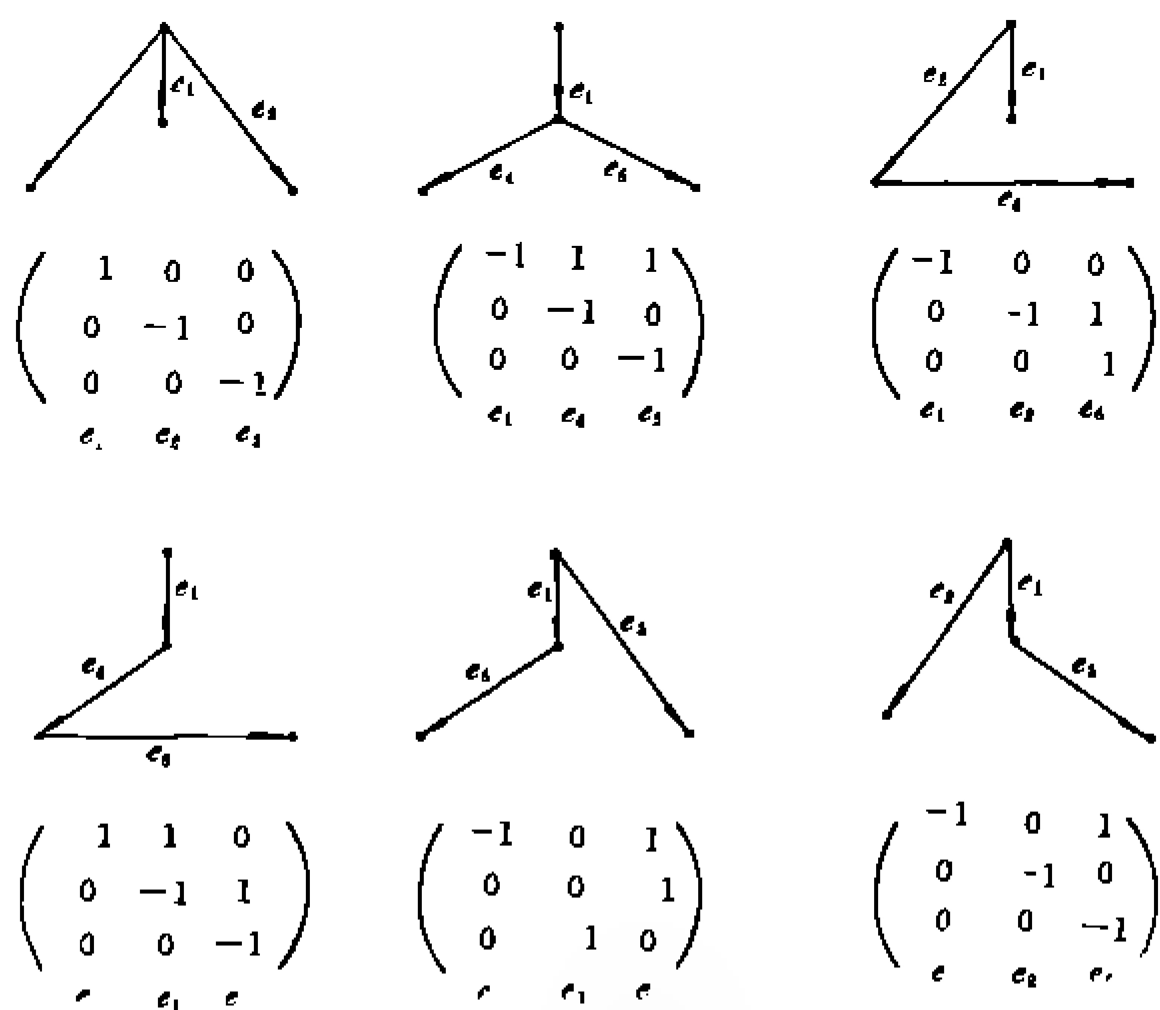


图 2-8 3

它们的值相等。根据 Binet Cauchy 定理可得下面的定理。

定理 以 v_k 为始点的外向树数目为：

$$\det(\vec{B}_k B_k^T)。$$

例：如图 2 8-2 所示，求以 v_1 为始点的外向树数目。

$$B_1 = \begin{pmatrix} -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix},$$

∴

$$\vec{B}_1 = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & -1 \end{pmatrix},$$

$$\vec{B}_1 B_1^T = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ -1 & 2 & 0 \\ 1 & -1 & 3 \end{pmatrix}。$$

所以，以 v_1 为始点的外向树数目为：

$$\det(\overrightarrow{B}_1 B_1^T) = 6。$$

类似的理由，若令以 v_k 为终点的基本关联矩阵 B_k 中(−1)元素改为零，用 $\overleftarrow{B}_k$ 表示这样取得的新矩阵。同样可得：

定理 以 v_k 为终点的内向树数目为

$$\det(\overleftarrow{B}_k B_k^T)。$$

例：如图 2 8 2 所示，求以 v_4 为终点的内向树数目。

$$B_4 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \end{bmatrix}, \quad \overleftarrow{B}_4 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\overleftarrow{B}_4 B_4^T = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 1 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 1 \end{bmatrix}。$$

故以 v_4 为终点的内向树数目为：

$$\det(\overleftarrow{B}_4 B_4^T) = 6。$$

这 6 个内向树的图象见图 2 8 4。

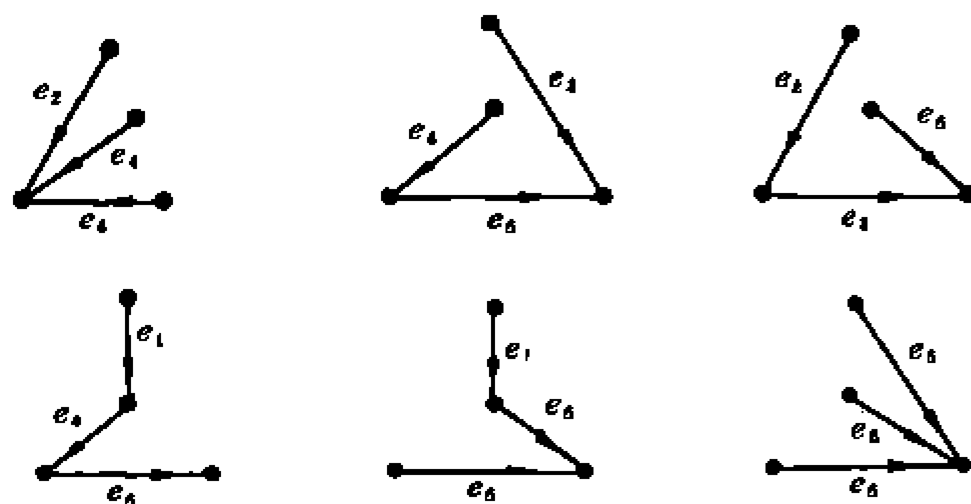


图 2-8 4

§9 二 元 树

二元树是一种特殊形式的树，也是比较重要的一种图。举例如下：

例 1：把 a, b, c 三个不同的数，按图 2-9-1 所示的框图使之按自然顺序从小到大排列。

例 2：如图 2 9-2 所示从一点向左下行为 0，右下行为 1。这样的树，它的“叶子”给出了一套编码如图 2-9-2 所示的{000，001，010，011，10，11}。用这些编码表达的信息很容易

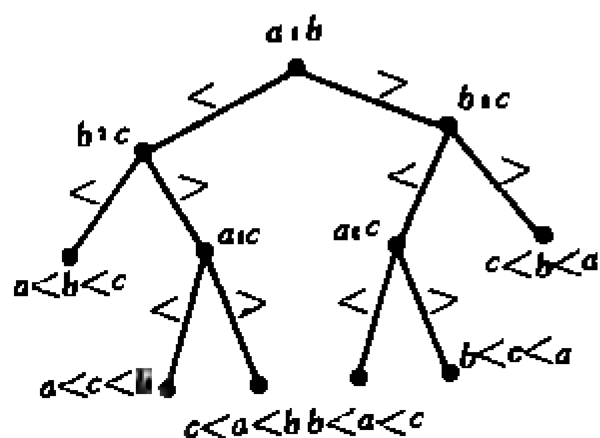


图 2-9-1

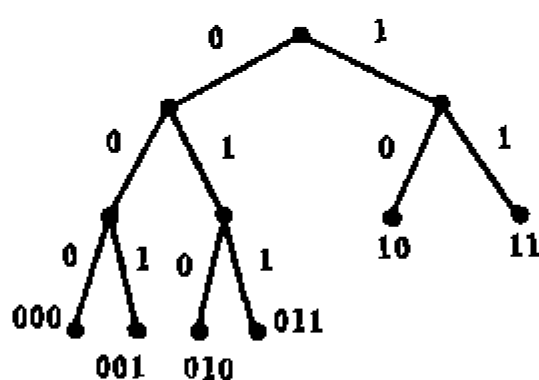


图 2-9-2

区别开来，不至含混不清。编码的要求在于用比较短的码来表示常用的符号，而少用的符号的码可以稍长一些，这样使得表达信息的码的长度尽可能短一些。

为什么说二元树的叶子给出的码容易识别呢？以上面的例子来说明如下：

例如，接收信息 01110010000011110，从树根开始下行走到叶子就算是一个码找出了，然后再从树根开始另一个码的搜索，不难分辨出它们是 011，10，010，000，001，11，10。

所谓二元树就是外向树的一种。外向树中入度为零的顶点叫做外向树的"源"，或叫做树根。其它各顶点的入度均为 1。二元树的特征是除了出度为零的叶子外，其它各顶点的出度均$\leqslant 2$。出度非零的顶点叫做分枝点。

v_i 是外向树的分枝点，若从 v_i 到 v_j 有一条有向边，则 v_j 称为是 v_i 的儿子或 v_i 是 v_j 的父亲。所谓二元树，即每一分枝点最多只有两个儿子。

二元树结构在计算机里容易表示。每一顶点用一单元图表示如下：

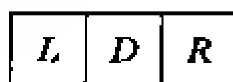


图 2-9-3

其中 D 是该顶点的讯息，L 为左指针，R 为右指针。这样，图 2-9-4 的树可以表示为图 2-9-5。

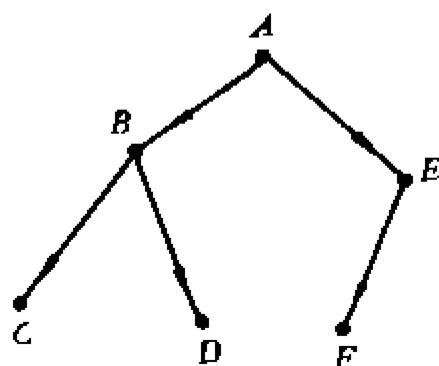


图 2-9-4

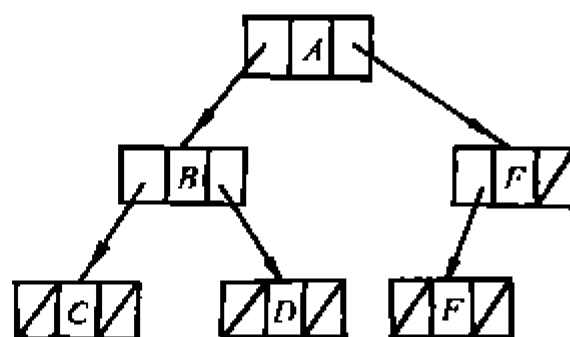


图 2-9-5

一般的外向树都可以转化为二元树来表示它。例如下面的外向树见图 2-9-6。可用二元树表示如图 2-9-7。

由上图可见一顶点左下方是它的"儿子"，右下方是它的"兄弟"。如果它有若干个儿子，只指出其中一个儿子，而其它儿子是这儿子的兄弟，依次类推。

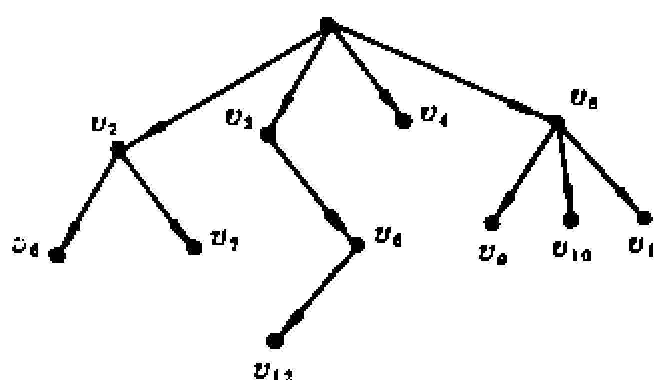


图 2-9-6

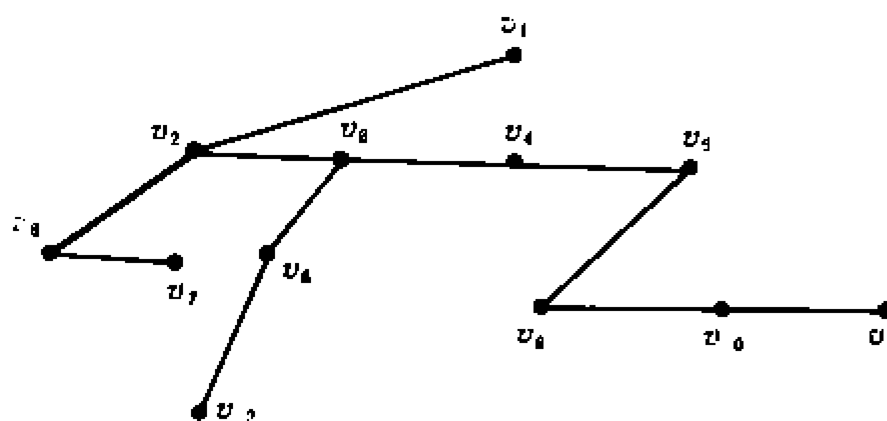


图 2-9-7

§10 Huffman 树

上节例 2 介绍了一棵二元树的叶子给出一组编码。比如英文二十六个字母出现的几率是不一样的，有的字母出现的次数多，如 e 和 t；有的出现的几率很少，如 z 和 q 等。在编码时，要求常见的字母码的长度短一些，如何设计最佳的编码？所谓最佳的标准是使得下面码长的数学期望值

$$L = \sum_{i=1}^{26} p_i l_i$$

达到最小。其中 l_i 是第 i 个字母的码的长度，而 p_i 是第 i 个字母出现的几率。

一组编码可对应一棵二元树，它的叶子 v_i 分别对应一权 p_i，v_i 到树根的长度（每条边的长度设为 1）为 l_i。因此设计最佳编码问题可归结为找一棵二元树，使得权

$$m(T) = \sum_{(i)} p_i l_i \text{。}$$

取最小值。其中 $\sum_{(i)}$ 是对所有的叶子求和。这样的树称为最佳二元树，又叫 Huffman 树。

假定：

$$p_1 \leqslant p_2 \leqslant \cdots \leqslant p_k \text{。}$$

首先证明，若 T 是带权 $p_1, p_2, \cdots, p_k$ 的最优二元树，则 p_1、p_2 必定是一对兄弟。因 p_1 最小，故对应的长度 l_1 应最大。同时 p_1 不可能没有兄弟。如若不然，如图 2-10-1，立即可

得权更小的树。这只要令它的“父亲”结点带权 p_1 即可。

图 2-10-1

设 T 是权为 $p_1,p_2,\cdots,p_k$ 的最优二元树，其中 p_1 和 p_2 是兄弟，而树 T_1 是令它们的“父亲”结点带权 p_1+p_2 后的二元树，如图 2-10-2 所示。显然有：

$$m(T_1)=m(T)-p_1-p_2$$

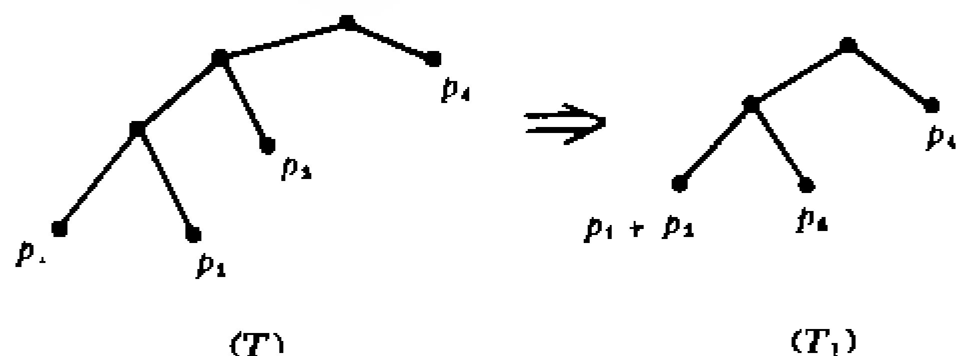


图 2-10-2

又令 $\overline{T}_1$ 是带权 $p_1+p_2,p_3,\cdots,p_k$ 的最优二元树。而 $\overline{T}$ 是图 2-10-3 左图 $\overline{T}_1$ 中叶子结点 p_1+p_2 向下延伸为分别带权 p_1 和 p_2 两个“儿子”的结点，即用图 2-10-3 的右图取代之所得结果。注意右图中新增加的树枝。

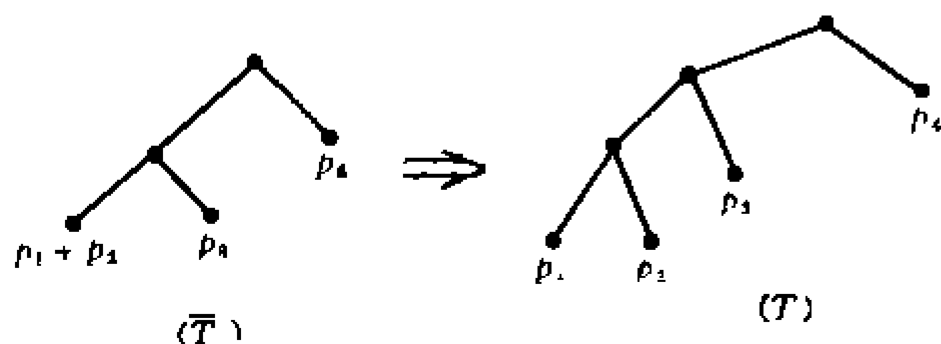


图 2-10-3

显然

$$m(\overline{T})=m(\overline{T}_1)+p_1+p_2,$$

∴
$$m(\overline{T}_1)=m(\overline{T})-p_1-p_2。$$

由于 $\overline{T}_1$ 是最优树，所以

$$m(\overline{T}_1)\leqslant m(T_1),$$

即

$$m(\overline{T})-p_1-p_2\leqslant m(T)-p_1-p_2,$$

故 $$m(\bar{T})\leqslant m(T)。$$

由于 T 是最优二元树，故

$$m(\bar{T}) = m(T)。$$

这就是说：可以从一带权 $p_1+p_2, p_3, \cdots, p_k$ 的最优二元树导出了一带权 $p_1, p_2, \cdots, p_k$ 的最优二元树。即从带 k 个权的最优二元树问题简化为带 $k-1$ 个权的最优二元树问题。依次类推最后导致带两个权的最优二元树问题。

例：(*a*) 求带权 0.01，0.03，0.03，0.03，0.05，0.05，0.2，0.6；

(*b*) 求带权 4，6，10，14，15，15，16，20

这两个问题的最优二元树。

算法说明：例(*a*)，图 2-10-4 中

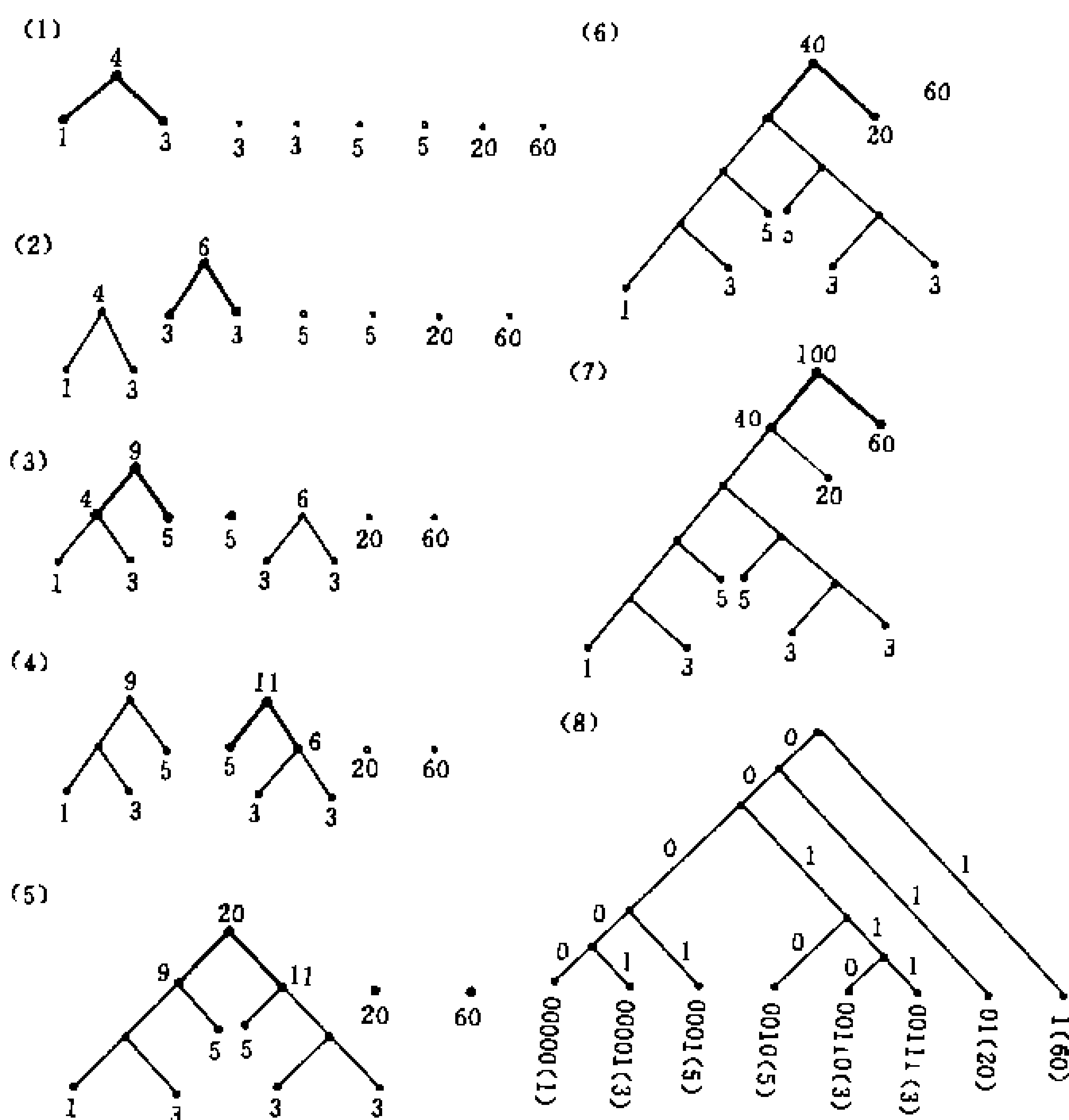


图 2-10-4

(1) 从 0.01，0.03，0.03，0.03，0.05，0.05，0.2，0.6 的带权最优二元树，导致找 0.03，0.03，0.04，0.05，0.05，0.2，0.6 的最优二元树；

(2) 从(1)的结果导致求带权 0.04，0.05，0.05，0.06，0.2，0.6 的最优二元树；

(3) 从(2)的结果导致求带权 0.05,0.06,0.09,0.2,0.6 的最优二元树，余此类推，详见图 2 10-4。为方便起见，所有的权都乘以 100，比如 0.01 便写为 1，0.60 便写为 60，还要特别说明了的图 2-10-4 中的(7)是最后结果，(8)和(7)从图的拓扑结构上是完全相同的，不过(8)是编码，例如叶片 01(20)，表示权为 0.20 的码为 01，依此类推。

例(b)和(a)相类似可得最优二元树如图 2-10-5 所示。

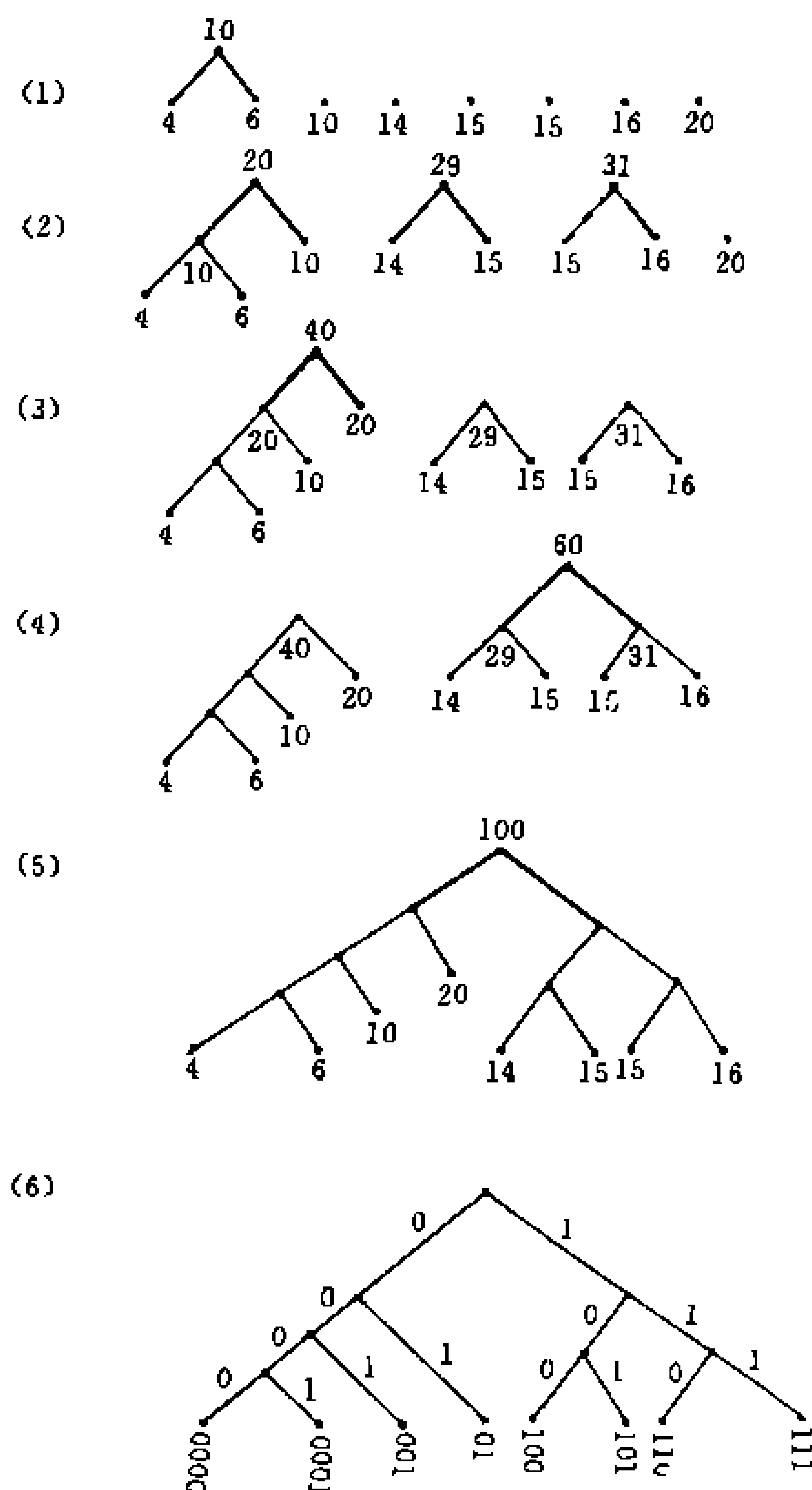


图 2-10-5

§11 搜 索 树

搜索树是实际中常用的一种方法。事实上前面已涉及到这个概念，现在只不过作为一种方法独立提出来。利用树的方法可以使得搜索过程中状态变化复杂的现象变得条理清

晰,从而找到最有效的方法。举例说明如下:

例 1: 若有 n 根火柴,甲、乙两人依次从中取走 1 根或 2 根,但不能不取。谁取走最后一根谁就是胜利者。

为了说明方法,不防设 $n=7$。在图 2-11-1 中用 $\boxed{7}$ 表示轮到甲取时,有 7 根火柴,④表示轮到乙取时有 4 根火柴。余此类推。

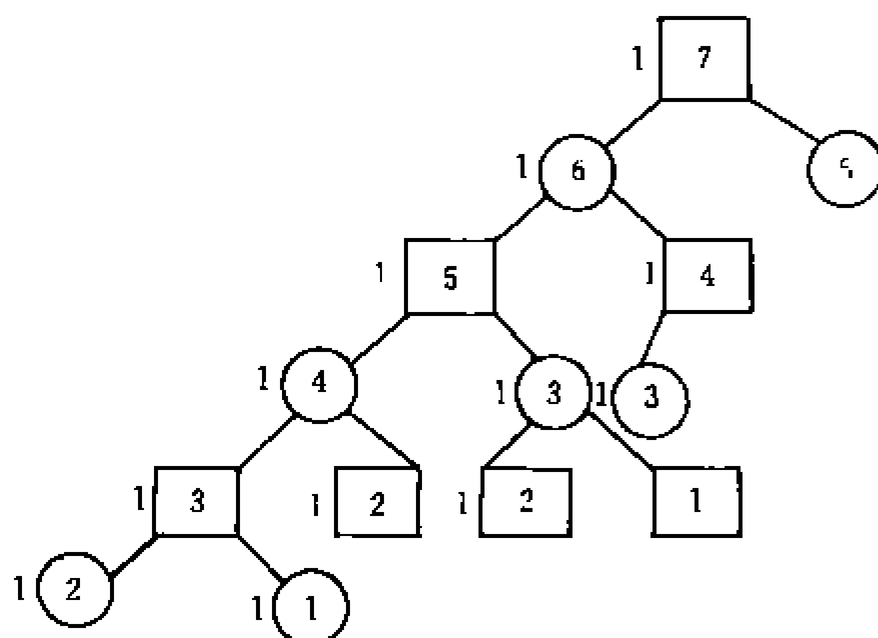


图 2-11-1

显然一当出现 $\boxed{1}$ 或 $\boxed{2}$ 状态,甲取胜,不必再搜索下去。同样①或②是乙取胜的状态。

若甲取胜时,设其得分为 1,乙取胜时甲的得分为 −1。无疑轮到甲作出判决时,他一定选择能取值 1 的对策;而轮到乙作出判决时,它将选取使甲失败,即选能取值 −1 的对策。这个道理是显而易见的。比如甲遇到图 2-11-2 的状态时,甲应选 $\max(1,-1)=1$,即甲应取 1 根火柴使状态进入③。同理乙遇到图 2-11-3 的状态时,乙应选取 $\min(-1,1)=-1$,使甲进入必然失败的状态 $\boxed{3}$ 为好。如图 2-11-1 所示,开始时若有 7 根火柴,先下手者胜局已定,除非对手失误,因⑥时取值 1。故 $\boxed{7}$ 取 1,而状态⑤的搜索可以省略去。即 $\boxed{7}$ 状态的甲决策使之进入⑥即可。这样达到剪枝的目的。各点的值是自下而上回溯的。

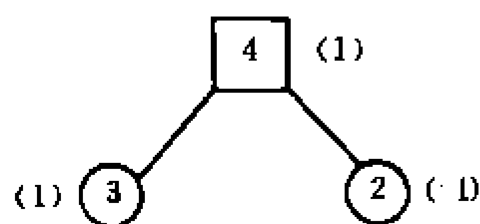


图 2-11-2

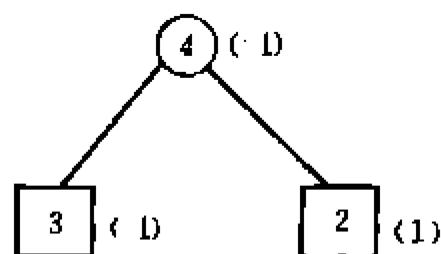


图 2-11-3

§12 流动商人问题与分支定界法

本章剩下的部分介绍几个应用,即解决实际问题的例子。

设 $v_1,v_2,\cdots,v_n$ 是已知的 n 个城市,若从某一城市 v_1 出发,经过各城市各一次最后返回 v_1,但要求路程最短。实际上就是求总长度最短的哈密尔顿(Hamilton)回路。

若对 n 个城市进行排列，若 v_i 到 v_j 的路长和 v_j 到 v_i 的相等时，结果将有 $\frac{1}{2}(n-1)!$ 种方案。由 Stirling 公式：

$$n! \approx \sqrt{2n\pi}\left(\frac{n}{e}\right)^n$$

随着 n 的增加，$n!$ 的值成指数型激剧增大。以 $n=20$ 为例，$20! \approx 2.4228\times10^{18}$，用每秒作 10^7 个方案比较的快速电子计算机也要 $2.4228\times10^{18}/(365\times24\times3600\times10^7)\approx2.4228\times10^{18}/(3.1536\times10^{14})\approx7.68\times10^3$ 年这是不现实的问题。于是它的能法便更引人注目了。下面介绍一种分枝定界方法。设 $n=5$，5 个城市间的距离用矩阵 $\boldsymbol{D}$ 表示如下：

$$\boldsymbol{D}=(d_{ij})_{5\times5}=\begin{bmatrix}\infty & 11 & 1 & 17 & 2\\ 11 & \infty & 23 & 1 & 3\\ 1 & 23 & \infty & 10 & 8\\ 17 & 1 & 10 & \infty & 6\\ 2 & 3 & 8 & 6 & \infty\end{bmatrix}\begin{matrix}v_1\\v_2\\v_3\\v_4\\v_5\end{matrix}$$

$$\quad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$$

其中 d_{ij} 表 v_i 到 v_j 的距离。上面矩阵是对称的，即从 v_i 到 v_j 的距离等于从 v_j 到 v_i 的距离。若是单行道即 $\boldsymbol{D}$ 为非对称情况要复杂得多了。

(1) 从五点中找出五条距离最短的边 v_1v_3，v_2v_4，v_2v_5，v_1v_5，v_4v_5 而且 $d_{13}+d_{24}+d_{25}+d_{15}+d_{45}=1+1+3+2+6=13$。

从图 2-12-1(a)可见这五条边不是 Hamilton 回路，因 v_5 点的度超过 2；

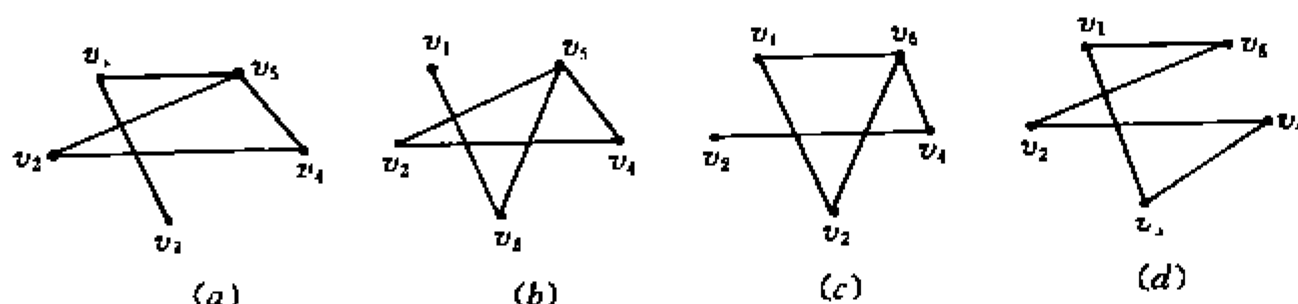


图 2-12-1

(2) 考虑排除 v_1v_5 边的情况下选取五条距离最短的边，以 v_3v_5 代替 v_1v_5 得：

$$d_{13}+d_{24}+d_{25}+d_{35}+d_{45}=1+1+3+8+6=19。$$

如图 2-12-1(b)，和(1)一样 v_5 的度依然为 3。

(3) 保持 v_1v_5 边得下界 13，而排除 v_1v_5 边的下界为 19，故继续沿着保持 v_1v_5 边的条件往下搜索，可得 $d_{13}+d_{15}+d_{24}+d_{35}+d_{45}=18$。如图 2-12-1($c$)依然不合理。

(4) 保存 v_1v_5、v_2v_5 保存 v_4v_5 是不可能，但排除 v_4v_5 得：$d_{13}+d_{15}+d_{24}+d_{25}+d_{35}=15$。如图 2-12-1($d$)，仍不合理。

(5) 在保持 v_1v_5、v_2v_5，排除 v_4v_5、v_3v_5 的条件下得：

$$d_{13}+d_{15}+d_{24}+d_{25}+d_{34}=17。$$

如图 2-12-1(e)所示得一 Hamilton 回路。

由于(2)、(3)所得的下界超过 17，故无搜索下去的价值。

把上面搜索过程用下面搜索树形式表示出来。其中$S^{(2)}(13、24、25、35、45)=19$表示搜索过程的第2步得$d_{13}+d_{24}+d_{25}+d_{35}+d_{45}=19$树枝注以(15)或$(\overline{15})$者分别表示确保$v_1v_5$边和排除$v_1v_5$边的意思。上面只用5次计算便得出结果：

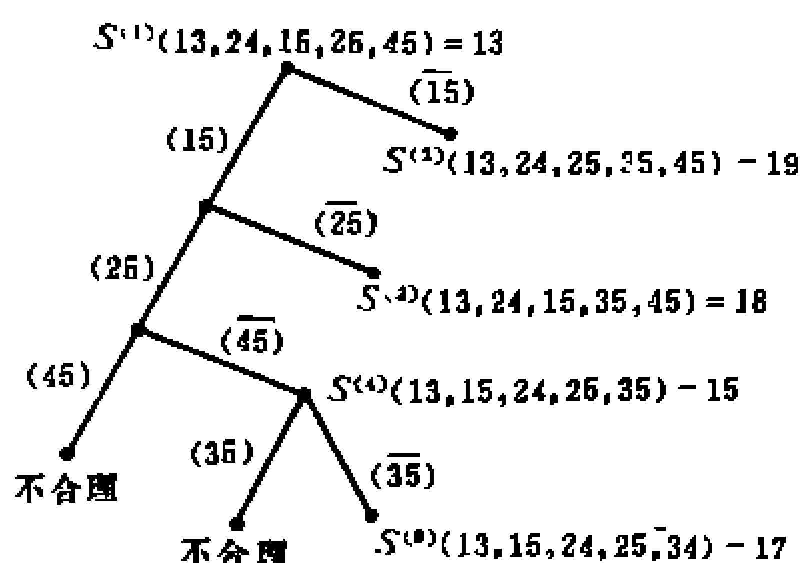


图 2-12-2

例：求下列流动商人问题的解：

$$D=\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix}\begin{pmatrix} \infty & 5 & 3 & 4 & 6 \\ 5 & \infty & 2 & 10 & 9 \\ 3 & 2 & \infty & 8 & 7 \\ 4 & 10 & 8 & \infty & 1 \\ 6 & 9 & 7 & 1 & \infty \end{pmatrix}$$
$$\qquad\qquad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$$

我们把搜索过程列表如图2-12-3；符号同上。

上面的例子中距离矩阵都是对称的，即i到j的旅程和j到i的旅程是相同的。现在介绍一种方法可以解非对称型的问题，即d_{ij}不一定等于d_{ji}。但对称型是它的特例，所以算法对对称型仍然有效。

$$D=\begin{pmatrix} \infty & 5 & 3 & 4 & 6 \\ 5 & \infty & 2 & 10 & 9 \\ 3 & 2 & \infty & 8 & 7 \\ 4 & 10 & 8 & \infty & 1 \\ 6 & 9 & 7 & 1 & \infty \end{pmatrix}\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix}$$
$$\qquad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$$

为了便于理解算法的思想，不妨将D看作是旅费矩阵。每行抽取最小元素，并令矩阵D每行的所有元素减去该行的最小元素得

$$D_1=\begin{pmatrix} \infty & 2 & 0 & 2 & 3 \\ 3 & \infty & 0 & 8 & 7 \\ 1 & 0 & \infty & 6 & 5 \\ 3 & 9 & 7 & \infty & 0 \\ 5 & 8 & 6 & 0 & \infty \end{pmatrix}\begin{matrix} 3 \\ 2 \\ 2 \\ 1 \\ 1 \end{matrix}$$

再用D_1各列的所有元素减去该列的最小元素得

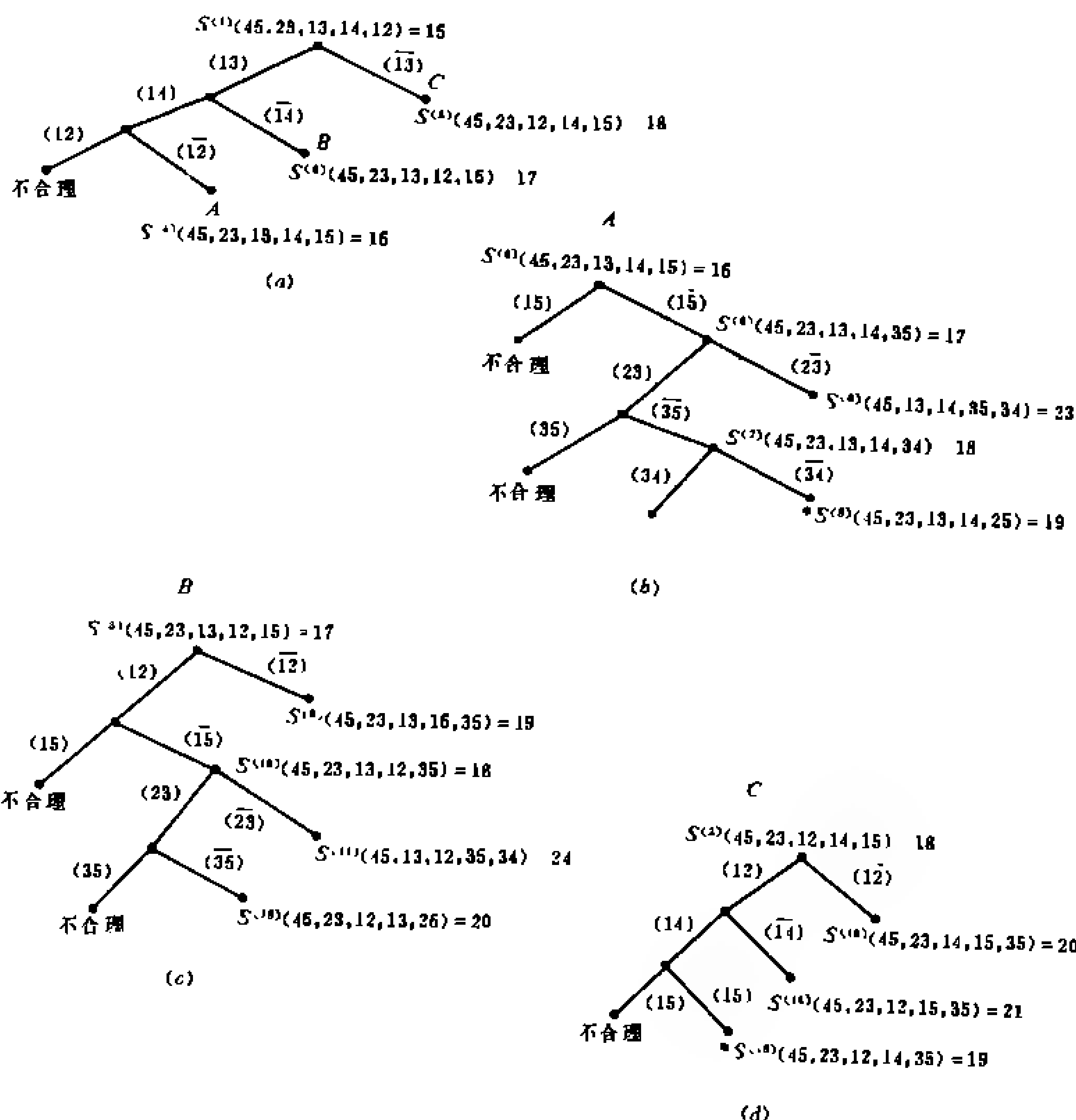


图 2 12 3

$$
D_2 = \begin{pmatrix}
\infty & 2 & 0 & 2 & 3 \\
2 & \infty & 0 & 8 & 7 \\
0 & 0 & \infty & 6 & 5 \\
2 & 9 & 7 & \infty & 0 \\
4 & 8 & 6 & 0 & \infty
\end{pmatrix} 10
$$

1

D_2 矩阵右下角 10—3＋2＋2＋1＋1＋1 的结果。

重要的结论在于以 D 为旅费矩阵的旅行商人问题的解和以 D_2 为旅费矩阵的解是一样的。

用每行的最小元素去减该行的所有元素，相当于从该行所对应的城市到其它城市的旅费一律降价，降的价是相同的。用每列的最小元素去减该列的所有元素，可看作是到该

列所对应城市的所有旅费一律降价，而且所降的价是相同的。旅行商人进入每个城市一次且仅一次，而且从该城市出去一次，也仅有一次。最佳的路径依然是降价后的最佳路径。这就证明了我们的结论，反正进出都一次。下面讨论 $\boldsymbol{D}_2$ 为旅费矩阵的问题的解。$\boldsymbol{D}_2$ 的特点是每行每列都有零元素至少一个。例如从 v_1 出发必然选择 v_3 作为下一站，因为 $d_{13}=0$。在 $\boldsymbol{D}_2$ 矩阵中划去 d_{13} 元素所在的行和列，并将 d_{31} 改为∞，这样做是为了避免出现 $v_1\to v_3\to v_1$ 的现象，这不符合问题的要求。消去第 1 行及第 3 列的原因在于每点进出仅一次。

$$\boldsymbol{D}_3=\begin{array}{c}v_2\\v_3\\v_4\\v_5\\\end{array}\begin{pmatrix}2&\infty&8&7\\\infty&0&6&5\\2&9&\infty&0\\4&8&0&\infty\end{pmatrix}10\text{。}$$
$$\qquad\quad v_1\quad v_2\quad v_4\quad v_5$$

在 $\boldsymbol{D}_3$ 矩阵中第一行无零元素出现，用最小元素去减该行所有元素得

$$\begin{array}{c}v_2\\v_3\\v_4\\v_5\\\end{array}\begin{pmatrix}0&\infty&6&5\\\infty&0&6&5\\2&9&\infty&0\\4&8&0&\infty\end{pmatrix}12=10+2\text{。}$$
$$v_1\quad v_2\quad v_4\quad v_5$$

v_3 的下一站必然是选 v_2 了，因 $d_{32}=0$，d_{32}所在的行和列去掉，d_{23}改为∞。同时 d_{21} 也改为∞，d_{23}改为∞是避免 $v_3\to v_2\to v_3$，但 d_{21} 改为∞是为了避免 $v_1\to v_3\to v_2\to v_1$，这也不符合旅行商人问题的要求，得

$$\boldsymbol{D}_4=\begin{array}{c}v_2\\v_4\\v_5\end{array}\begin{pmatrix}\infty&6&5\\2&\infty&0\\4&0&\infty\end{pmatrix}\text{。}$$
$$\qquad v_1\quad v_4\quad v_5$$

同样用 $\boldsymbol{D}_4$ 的第 1 行最小元素 5 去减第 1 行，用第 1 列的最小元素 2 去减第 1 列得

$$\boldsymbol{D}_5=\begin{array}{c}v_2\\v_4\\v_5\end{array}\begin{pmatrix}\infty&1&0\\0&\infty&0\\2&0&\infty\end{pmatrix}19=12+5+2\text{。}$$
$$\qquad v_1\quad v_4\quad v_5$$

v_2 的下一站自然选 v_5。依上述办法得

$$\boldsymbol{D}_6=\begin{array}{c}v_4\\v_5\end{array}\begin{pmatrix}0&\infty\\\infty&0\end{pmatrix}19\text{。}$$
$$\qquad v_1\quad v_4$$

故得一条路径

$$v_1\to v_3\to v_2\to v_5\to v_4\to v_1$$

旅费为 19，是否为最佳呢？以开始时 $v_1\to v_3$ 为例，若不取 $v_1\to v_3$ 究竟如何呢？若封锁 $v_1\to v_3$，在 $\boldsymbol{D}_2$ 中令 $d_{13}=\infty$，可得

$$
D_7 = \begin{pmatrix}
\infty & 2 & \infty & 2 & 3 \\
2 & \infty & 0 & 8 & 7 \\
0 & 0 & \infty & 6 & 5 \\
2 & 9 & 7 & \infty & 0 \\
4 & 8 & 6 & 0 & \infty
\end{pmatrix} 10 。
$$

用第 1 行的最小元素 2 去减该行其它元素得

$$
\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix}
\begin{pmatrix}
\infty & 0 & \infty & 0 & 1 \\
2 & \infty & 0 & 8 & 7 \\
0 & 0 & \infty & 6 & 5 \\
2 & 9 & 7 & \infty & 0 \\
4 & 8 & 6 & 0 & \infty
\end{pmatrix} 12 = 10 + 2 。
$$

12 是它的界，即封锁 $v_1 \to v_3$ 后旅费的下界。

现将搜索树表示如图 2.12.4：

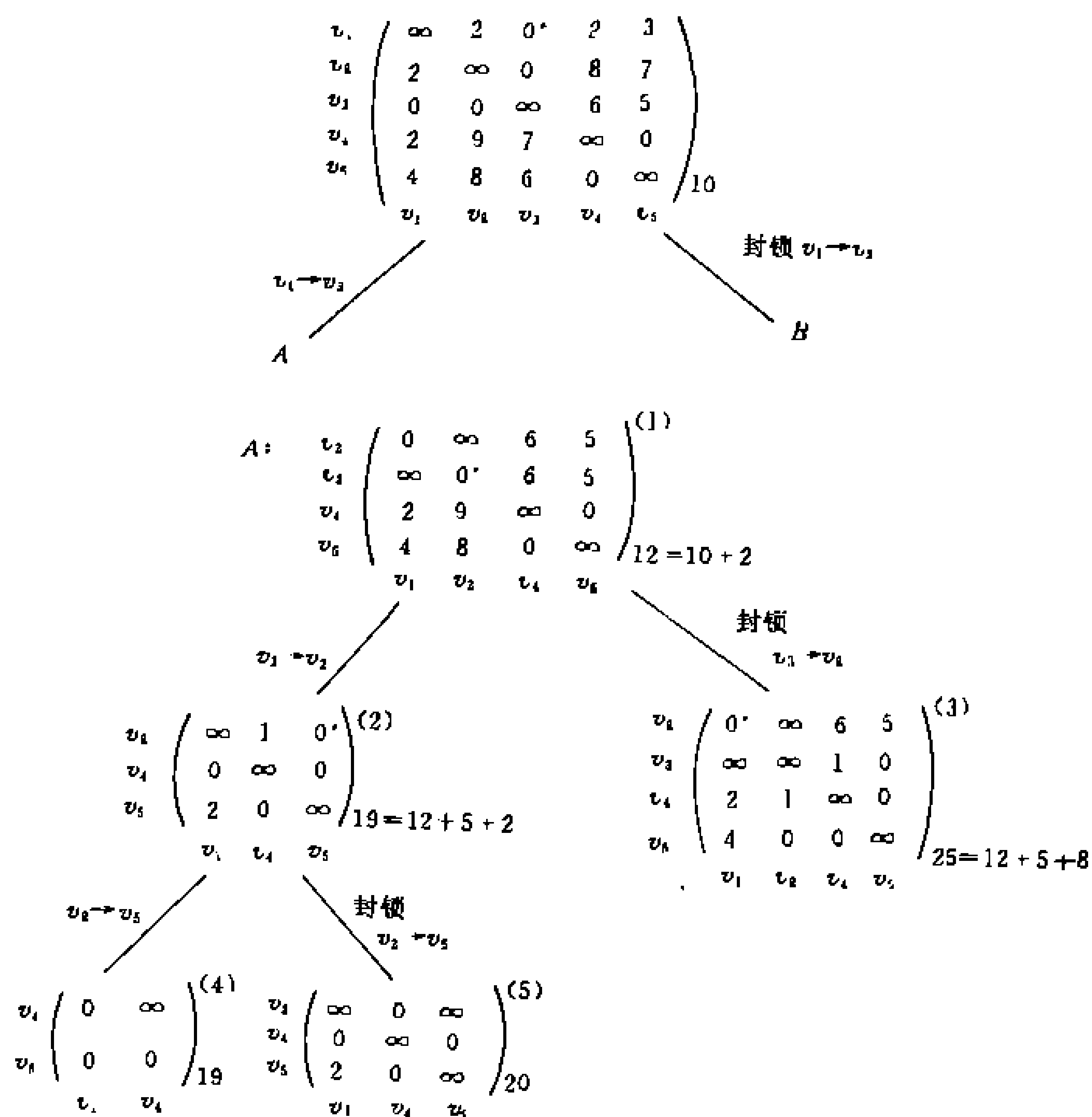


图 2.12.4(A)

$$B:\quad (8)\quad \begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline v_1 & \infty & 0 & \infty & 0 & 1 \\ v_2 & 2 & \infty & 0 & 8 & 7 \\ v_3 & 0 & 0^* & \infty & 6 & 5 \\ v_4 & 2 & 9 & 7 & \infty & 0 \\ v_5 & 4 & 8 & 6 & 0 & \infty \end{array}\quad 12$$

$v_3\to v_2$:

$$(9)\quad \begin{array}{c|cccc} & v_1 & v_3 & v_4 & v_5 \\ \hline v_1 & \infty & \infty & 0 & 1 \\ v_2 & 0 & \infty & 8 & 7 \\ v_4 & 0 & 1 & \infty & 0 \\ v_5 & 2 & 0 & 0 & \infty \end{array}\quad 20>19$$

封锁 $v_3\to v_2$:

$$(10)\quad \begin{pmatrix} \infty & 0 & \infty & 0^* & 1 \\ 2 & \infty & 0 & 8 & 7 \\ 0 & \infty & \infty & 6 & 5 \\ 2 & 9 & 7 & \infty & 0 \\ 4 & 8 & 6 & 0 & \infty \end{pmatrix}\quad 12$$

$v_1\to v_4$:

$$(11)\quad \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_5 \\ \hline v_2 & 2 & \infty & 0 & 7 \\ v_3 & 0 & \infty & \infty & 5 \\ v_4 & 2 & 1 & 7 & 0 \\ v_5 & 4 & 0 & 6 & 0 \end{array}\quad 20>19$$

封锁 $v_1\to v_4$:

$$(12)\quad \begin{pmatrix} \infty & 0 & \infty & \infty & 1 \\ 2 & \infty & 0 & 8 & 7 \\ 0 & \infty & \infty & 6 & 5 \\ 2 & 9 & 7 & \infty & 0^* \\ 4 & 8 & 6 & 0 & \infty \end{pmatrix}\quad 12$$

$v_4\to v_5$:

$$(13)\quad \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & \infty & 0 & \infty & \infty \\ v_2 & 2 & \infty & 0 & 2 \\ v_3 & 0 & \infty & \infty & 0 \\ v_5 & 0 & 4 & 2 & \infty \end{array}\quad 22$$

封锁 $v_4\to v_5$:

$$C:\quad (14)\quad \begin{pmatrix} \infty & 0 & \infty & \infty & 0 \\ 2 & \infty & 0 & 8 & 6 \\ 0 & \infty & \infty & 6 & 4 \\ 0 & 7 & 5 & \infty & \infty \\ 4 & 8 & 6 & 0 & \infty \end{pmatrix}\quad 15$$

图 2-12-4(B)

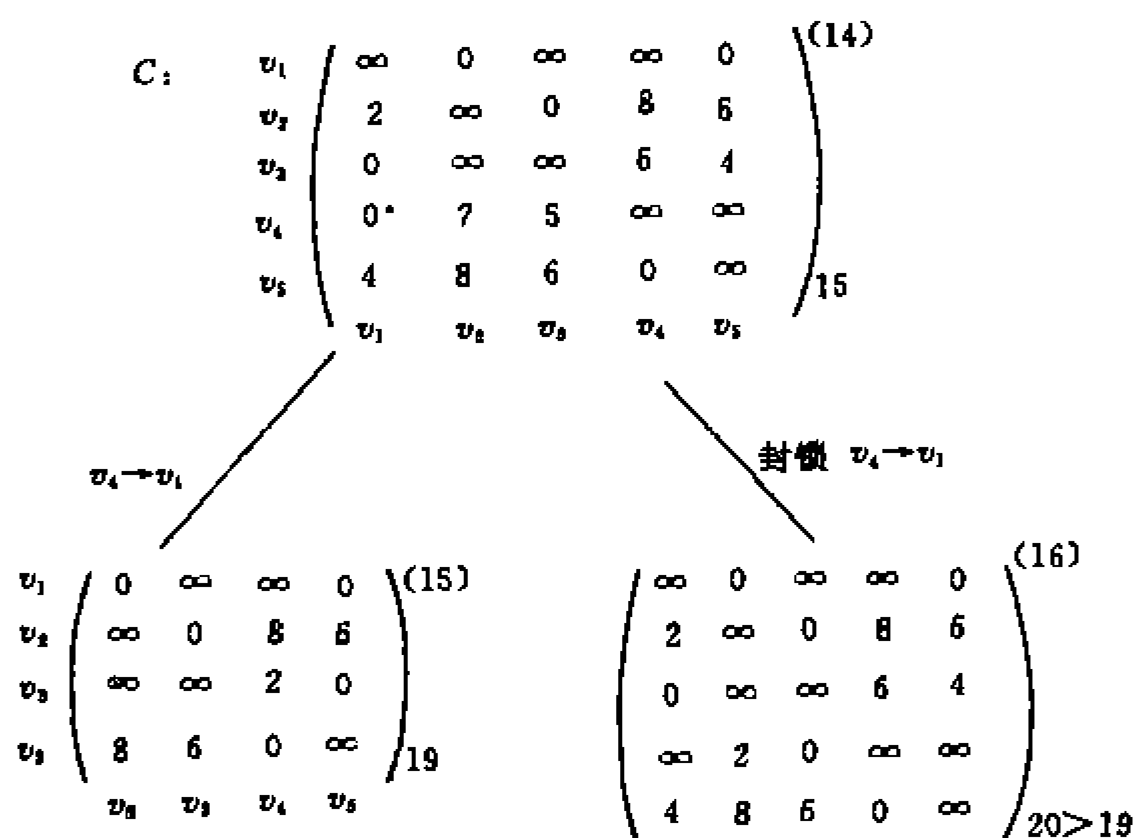


图 2-12-4(C)

故得最佳路径：

$$v_1 \to v_3 \to v_2 \to v_5 \to v_4 \to v_1,$$

总费用为 19，矩阵右上肩里的数是搜索的顺序。例如右上肩(5)便是・(5)对应的矩阵估的界为 20>19，故停止继续搜索。其它如(6)，(7)，(9)，…，(13)，(15)，(16)，都是由于估界超过合格的 19，无搜索的价值而放弃。

§13 最佳匹配问题

匹配问题是运筹学的重要问题之一，也是图论的重要内容，它在所谓“人员分配问题”和“最优分配问题”中有重要应用。本书举例说明如何利用分枝定界法求问题的解。

已知用 A,B,C,D 四种材料，生产Ⅰ，Ⅱ，Ⅲ，Ⅳ四种产品的成本如下面矩阵所表示：

$$E = \begin{array}{c} A \\ B \\ C \\ D \\ \\ \end{array}\!\!\begin{array}{c} \begin{pmatrix} 99 & 6 & 59 & 73 \\ 79 & 15 & 93 & 87 \\ 67 & 93 & 13 & 81 \\ 16 & 79 & 86 & 26 \end{pmatrix} \\ \begin{matrix} \text{Ⅰ} & \text{Ⅱ} & \text{Ⅲ} & \text{Ⅳ} \end{matrix} \end{array}$$

问题是哪种方案使成本最低？假定一种材料用作某种产品就不再用作其它产品的原料。

步骤如下：

(1) 确定材料 A 用作产品Ⅰ的原料。在矩阵 E 中去掉第一行、第一列后得矩阵：

$$\begin{array}{c} B \\ C \\ D \\ \\ \end{array}\!\!\begin{array}{c} \begin{pmatrix} 15 & 93 & 87 \\ 93 & 13 & 81 \\ 79 & 86 & 26 \end{pmatrix} \\ \begin{matrix} \text{Ⅱ} & \text{Ⅲ} & \text{Ⅳ} \end{matrix} \end{array}$$

即排除 A 作为其它产品原料的条件下分别寻找产品Ⅱ，Ⅲ，Ⅳ的材料。结果选得 B 作为Ⅱ的原料，C 作为Ⅲ的原料，D 作为Ⅳ的原料，用

$$\begin{pmatrix} \underline{A}\ B\ C\ D \\ 153 \end{pmatrix}$$

来表示。其中 153 是 $e_{11}+e_{22}+e_{33}+e_{44}=99+15+13+26$ 的结果。符号 $\underline{A}B\ C\ D$ 表示 A,B,C,D 依次作为Ⅰ，Ⅱ，Ⅲ，Ⅳ的原料，$\underline{A}$ 表示在确定以 A 作为Ⅰ的原料是先决条件。只要 A,B,C,D 中没有重复出现的现象就算是合理的匹配。

同理以 B 作为Ⅰ的原料的条件下，得

$$\begin{pmatrix} \underline{B}\ A\ C\ D \\ 124 \end{pmatrix}$$

显然也是合理的，而且成本比 153 少。

确定 C 作为Ⅰ的原料和 D 作为Ⅰ的原料分别得

$$\begin{pmatrix} \underline{C}\ A\ A\ D \\ 158 \end{pmatrix},\begin{pmatrix} \underline{D}\ A\ C\ A \\ 108 \end{pmatrix}.$$

显然这两个选择都是不合理的。特别是

$$\begin{pmatrix} C\ A\ A\ D \\ 158 \end{pmatrix}$$

没有搜索的价值。因为在确定 C 作为Ⅰ的原料的条件下成本最少为 158，比已有的合理方案 $\begin{pmatrix} \underline{B}\ A\ C\ D \\ 124 \end{pmatrix}$ 成本高。

$$\begin{pmatrix} \underline{D}\ A\ C\ A \\ 108 \end{pmatrix}$$

说明以 D 做为Ⅰ的原料的条件下，成本的下界为 108，最有继续探索的价值。

(2) 在确定 D 作为Ⅰ的原料的条件下，分别以 A,B,C 作为Ⅱ的原料的条件下得

$$\begin{pmatrix} \underline{D\ A}\ C\ C \\ 116 \end{pmatrix},\begin{pmatrix} \underline{D\ B}\ C\ A \\ 117 \end{pmatrix},\begin{pmatrix} D\ C\ A\ A \\ 242 \end{pmatrix}。$$

显然 $\begin{pmatrix} \underline{D\ B}\ C\ A \\ 117 \end{pmatrix}$ 是合理的，而且比 $\begin{pmatrix} \underline{B}\ A\ C\ D \\ 124 \end{pmatrix}$ 更佳，是搜索到此所得的最好结果。$\begin{pmatrix} \underline{D\ C}\ A\ A \\ 242 \end{pmatrix}$ 是毫无搜索的价值，但 $\begin{pmatrix} D\ A\ C\ C \\ 116 \end{pmatrix}$ 由于下界 116 比最好结果 117 小，还要继续搜索。

(3) 在以 D,A 分别作为Ⅰ，Ⅱ的原料的条件下，令 B,C 分别为Ⅲ的原料得

$$\begin{pmatrix} \underline{D\ A\ B}\ C \\ 196 \end{pmatrix},\begin{pmatrix} \underline{D\ A\ C}\ B \\ 122 \end{pmatrix}$$

虽然都是合理方案，但由于成本高于 117，故最佳方案仍是

$$\begin{pmatrix} D\ B\ C\ A \\ 117 \end{pmatrix}$$

搜索过程如图 2-13-1 所示：

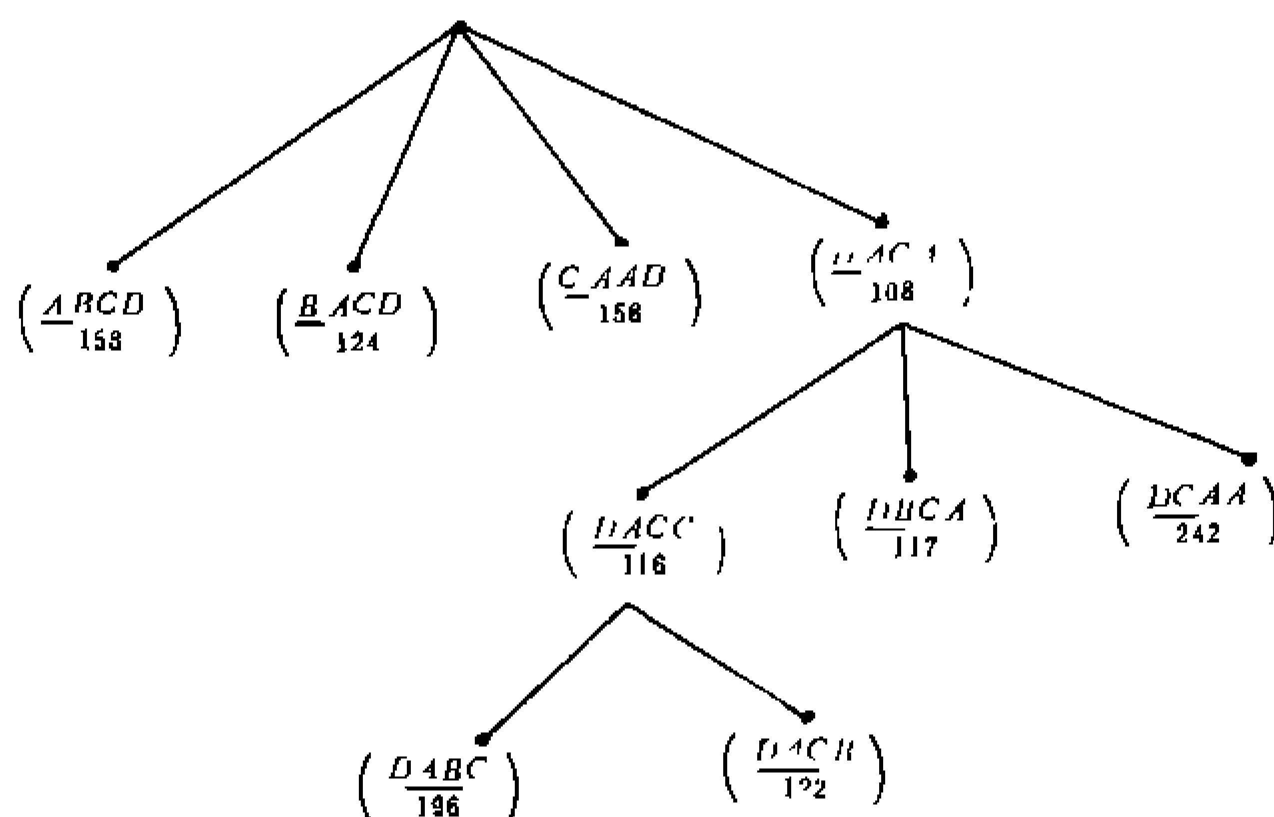


图 2-13-1

习　题

1. 试作C_5H_{12}的枝链结构图。
2. 图G是n个顶点m条边的图。若

$$A=(a_{ij})_{n\times n}\qquad a_{ij}=\begin{cases}1,\text{若 } v_i \text{ 与 } v_j \text{ 点相邻;}\\0,\text{其它。}\end{cases}$$

$$B=(b_{ij})_{n\times m}\qquad b_{ij}=\begin{cases}1,\text{若 } v_i \text{ 是 } e_j \text{ 的端点;}\\0,\text{其它。}\end{cases}$$

$$D=(d_{ij})_{n\times n}\qquad d_{ij}=\begin{cases}d(v_i),j=i\\0\quad ,j\neq i。\end{cases}$$

试证：　　$BB^T=A+D$。

3. 试证若图G的任意两点之间的道路是唯一的，则图G是树。
4. 试证若图G有n个顶点，$n-1$条边，不存在回路，则图G是连通图。
5. 求下列图的树的数目，并画出所有的树。
6. 利用分支定界法解下面的流动商人问题，其中D是6个顶点v_1,v_2,v_3,v_4,v_5间的距离矩阵。

$$D=\begin{matrix}v_1\\v_2\\v_3\\v_4\\v_5\end{matrix}\begin{pmatrix}0&34&2&50&59\\34&0&36&68&67\\2&36&0&51&60\\50&68&51&0&13\\59&67&60&13&0\end{pmatrix}$$
$$\qquad\quad v_1\quad v_2\quad v_3\quad v_4\quad v_5$$

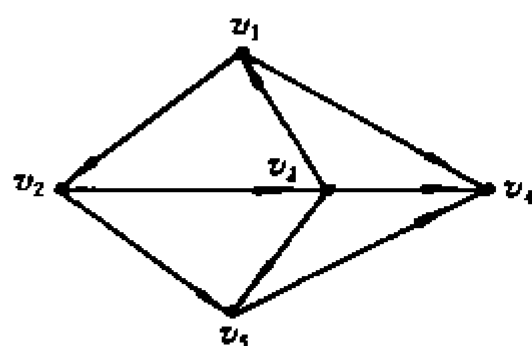


图　习题5

7. 用分支定界法求下面的最佳匹配问题。其中C是成本矩阵，A,B,C,D是材料，Ⅰ，Ⅱ，Ⅲ，Ⅳ是产品。

$$C=\begin{matrix}A\\B\\C\\D\end{matrix}\begin{pmatrix}5&7&7&3\\5&6&5&2\\7&6&6&3\\9&8&10&6\end{pmatrix}。$$
$$\qquad\quad \text{Ⅰ}\quad \text{Ⅱ}\quad \text{Ⅲ}\quad \text{Ⅳ}$$

8. 试利用Binet-Cauchy公式讨论内向树的数目为什么不是$det(\bar{B}_k\bar{B}_k^T)$。
9. 若已知n个字符的频率：

$$f_1\leqslant f_2\leqslant\cdots\leqslant f_n,$$

试写构造它们的Huffman码的程序。

10. 过给标号的n个顶点的树有n^{n-2}个。试证明之。（提示：实际上是n个顶点完全图V_n的支撑树数目）。
11. e是连通图G的一条边，若从G中消去e，结果$G\setminus\{e\}$子图为非连通图，则称e是G的桥，试证e是桥的充要条件是所有支撑树都包含e边。

12. 若图 G 的支撑树是唯一的，试证 G 本身就是树。

13. 设 G 是一连通图，C 是 G 的回路，试证的任一棵支撑树的补图（见第一章习题第16题）必然至少有 C 的一条边。

14. 求下图的中国邮路问题的解。

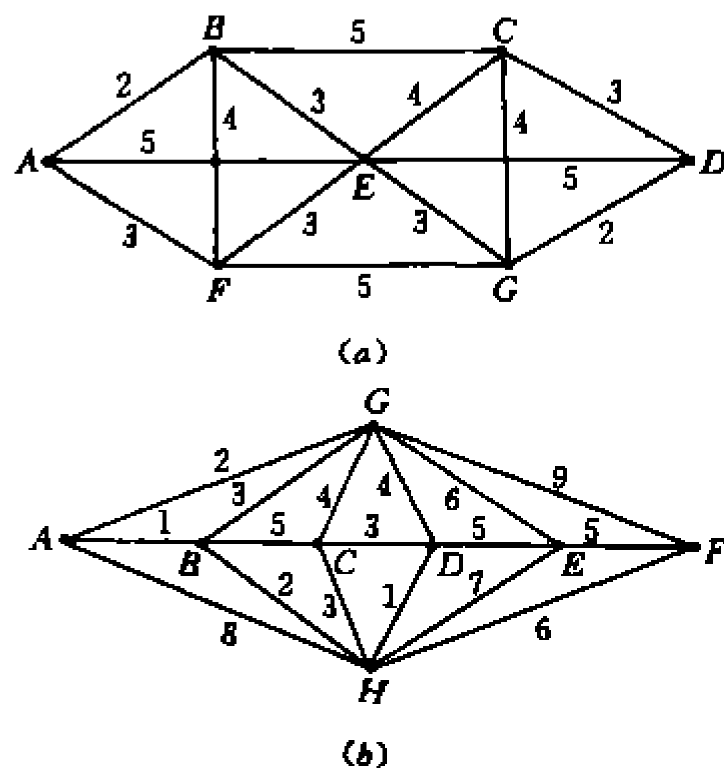


图　习题14

15. 设 T 是 n 个顶点图，v 是 T 上度数最大的点。

(a) 试证 $d(v)=2$ 时 T 是一条道路。

(b) 若 $d(v)=n-1$，试讨论 T 是一星图。

16. 试证下图存在一条哈密尔顿回路。

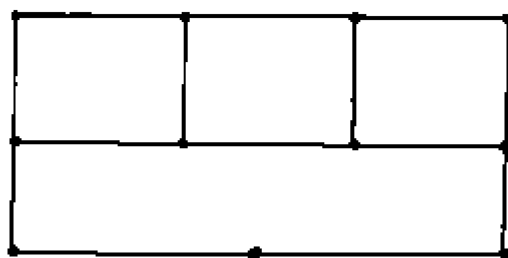


图　习题16

17. 试证下图存在一哈密尔顿道路，但不存在哈密尔顿回路。

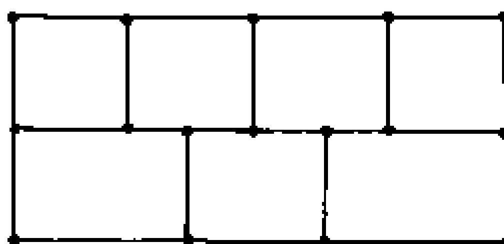


图　习题17

第三章　图 的 算 法

§1　最佳路径问题及其算法

一、设图 $G=(V,E)$，对 G 的每一条边 (v_i,v_j) 赋以权，G 连同在它边上的权称为赋权图

若一条边 (v_i,v_j) 的权表示它的长度，一条道路 $\mu=\{e_1,e_2,e_3,\cdots,e_k\}$ 的长度即为 μ 上所有边的长度的和。在赋权图中给定一个顶点 v_i（称为始点）及顶点 v_j（称为终点），所谓最短路径问题，就是在 v_i 和 v_j 之间的所有路径中，寻求长度最小的路径，这样的路径称为从 v_i 到 v_j 的最佳路径。

许多选最优的问题都和图论中的最佳路径问题相关。最佳路径问题在运筹学中有着许多实际意义。

比如图 3-1-1(a)是各城市间的交通网，求 v_0 点到其它各点的最短路径。各边的权可以是路径的长度，也可以是交通费用。

求一点到其它各点最短路径算法之一：

(1) 任取一棵以 v_0 为根的外向树。并求各点与 v_0 的距离 l_i，如图 3-1-1(b)所示，在各点所注的数值便是 l_i 的长度。显然它是与树的选择有关，是沿树枝回溯到 v_0 的结果。

(2) 树枝以外的边 v_iv_j 的长度设为 d_{ij}，若对于所有的 v_iv_j 边的两端点 v_i,v_j 不等式

$$l_j \leqslant l_i + d_{ij}$$

都成立，则这棵树便是所求。否则若对其中一对相邻顶点 v_i,v_j 有：

$$l_j > l_i + d_{ij}$$

则去掉以 v_j 为终点的树枝，代以 v_iv_j 边。修改由此引起路径长度的变化。如此反复进行，直到对树以外的边都进行了比较，无任何变化为止。这样所得的树即为所求。如图 3-1-1(b)所示。

二、求一点到其它各点最短路径算法之二——戴克斯徒拉(Dijkstra)算法

如若 $v_1v_2\cdots v_{n-1}v_n$ 是从 v_1 到 v_n 的最短路径，则 $v_1v_2\cdots v_{n-1}$ 也必然是从 v_1 到 v_{n-1} 的最短路径。根据这原理，前一节的问题也可按如下进行求解。

(1) 与 v_0 相邻的顶点有 v_1,v_5 两点，v_1 最接近于 v_0 点，联 v_0v_1；而且令 $l_1=1$。

(2) 和 $S=\{v_0,v_1\}$ 两点相邻接的点有 v_2,v_6,v_5，而且

$$l_2=2+l_1=3,$$

$$l_5=2,$$

$$l_6=l_1+d_{16}=3,$$

$$\min\{l_2,l_5,l_6\}=l_5=2。$$

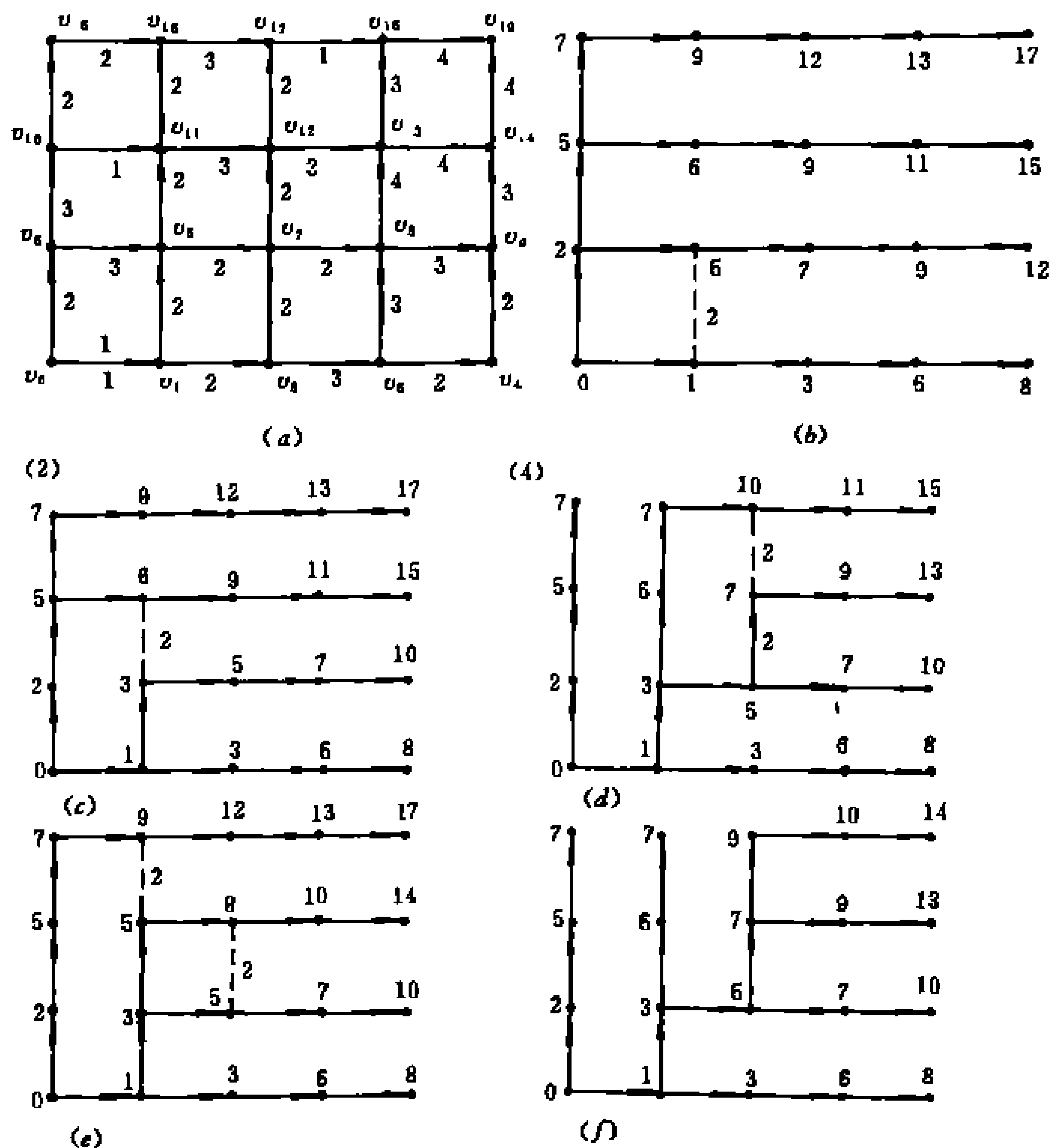


图 3 1-1

故连接 v_0v_5。

(3) 与 $S=\{v_0,v_1,v_5\}$ 相邻接的点有 v_2,v_6,v_{10}。

$$l_2=l_1+d_{12}=3,$$

$$l_6=\min\{l_1+d_{16},\ l_5+d_{56}\}=3,$$

$$l_{10}=l_5+d_{5,10}=5,$$

$$\min\{l_2,l_6,l_{10}\}=l_2=l_6=3。$$

故连接 v_1v_2，v_1v_6。

(4) 与 $S=\{v_0,v_1,v_2,v_5,v_6\}$ 相邻接的点有 v_3,v_7,v_{10},v_1 其中

$$l_3=l_2+d_{23}=6,$$

$$l_7=\min\{l_6+d_{67},\ l_2+d_{27}\}=\min\{5,5\}=5,$$

$$l_{10}=l_5+d_{5,10}=2+3=5,$$

$$l_{11}=l_6+d_{6,11}=3+2=5.$$

故联 v_2v_3，v_6v_7，v_6v_{11} 等等。

如此依次反复进行直到所有的顶点都连接起来为止，如图 3 1-2 所示。

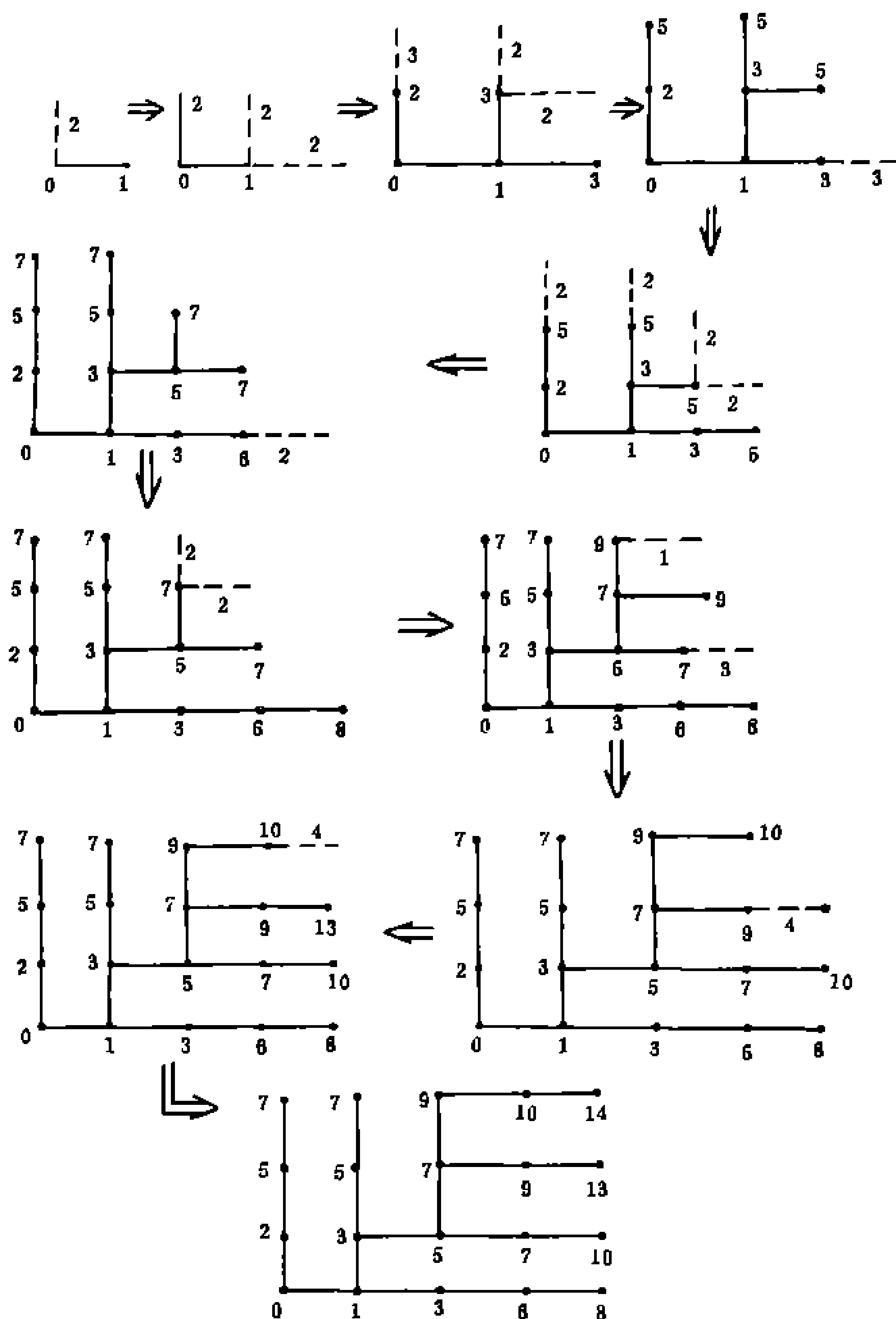


图 3 1-2

三、Dijkstra 算法步骤

(1) $S\leftarrow\{v_0\}$，$l(v_0)\leftarrow 0$，$l(v_i)\leftarrow\infty$，$i=1,2,\cdots,n$；　$i\leftarrow 0$，$\overline{S}\leftarrow V\setminus\{v_0\}$；

(2) 若 $\overline{S}$ 是空集则打印 S 后停止，否则转(3)；

(3) 对 $v_i\in\overline{S}$ 的所有点计算 $l(v_i)=\min\limits_{v_j\in S}\{l(v_i),\ l(v_j)+d_{ij}\}$；

(4) 令 $l(v_{i+1})=\min\limits_{v\in\overline{S}}\{l(v)\}$，$S\leftarrow S\cup\{v_{i+1}\}$，$\overline{S}\leftarrow\overline{S}\setminus\{v_{i+1}\}$；$i\leftarrow i+1$；转(2)。

其流程图(见图 3 1 3)。

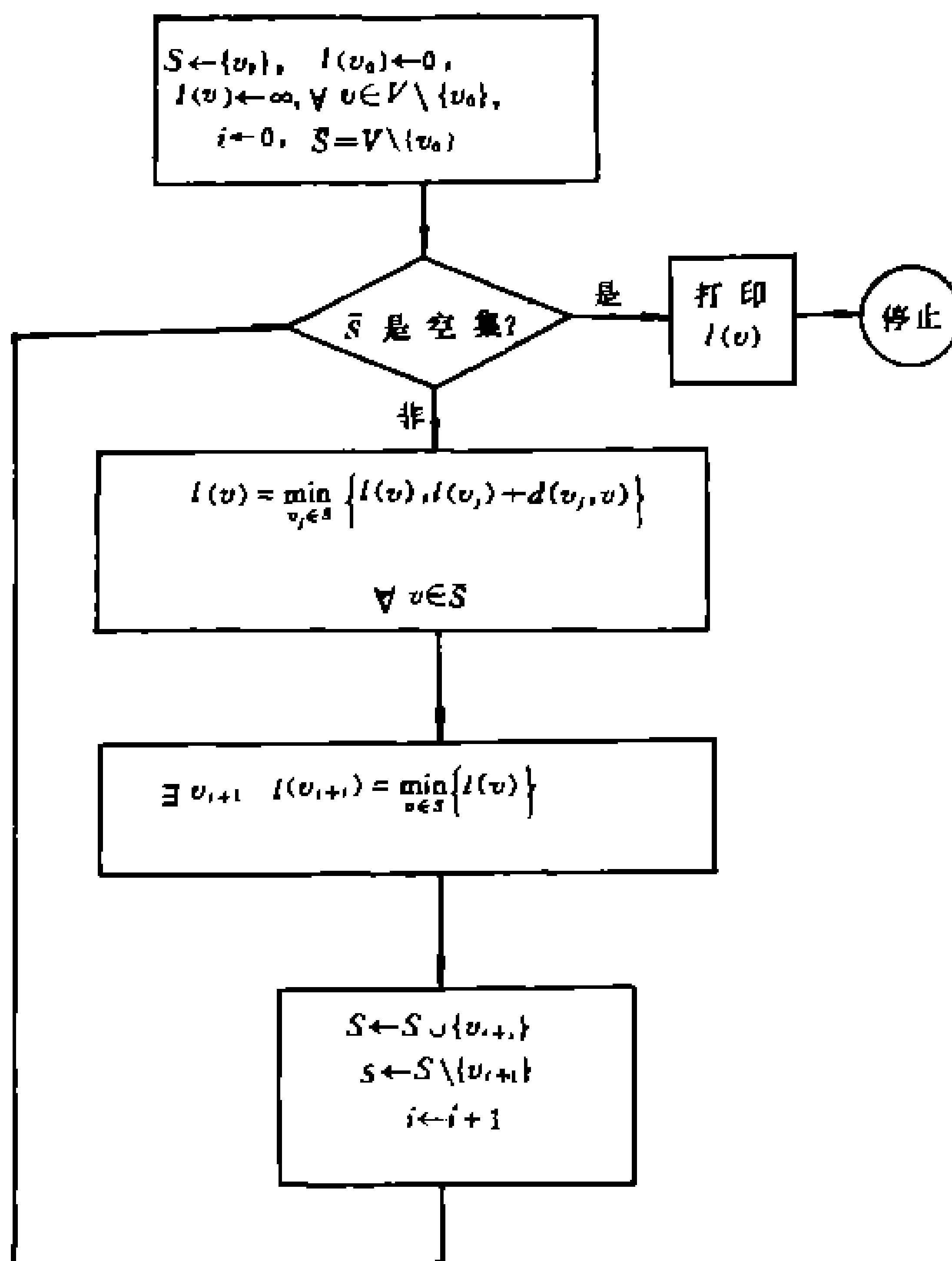


图　3-1-3

例：如若某一设备今后五年内的价格依次为100，110，115，120，130(单位)，已知该设备在第1，2，3，4，5 年内的维修费用分别为 10，15，20，30，60 单位，使用1，2，3，4，5 年后的折旧费为 50，40，30，20，0 单位。试问应如何更新该设备，使得到第 5 年终了时总费用最少?

问题导到求下图 3 1 4 中从 v_1 到 v_6 的最短距离：v_i 点表示第 i 点开始的状态，$(v_i$，

v_j)边上的数表示从第 i 年开始购买该设备到第 j 年开始的总费用。例如(v_1, v_2)边上的数为 $60=100+10-50$，即购置费 100 单位，第一年维修费用 10 单位，1 年后的机器折旧卖出 50 单位。又如(v_1, v_6)边上数为 $100+10+15+20+30+60=235$，即购买设备费 100 单位，5 年内的维修费用为 $10+15+20+30+40=115$，最后收回折旧费 0 单位，故总费用 215 单位，余此类推。

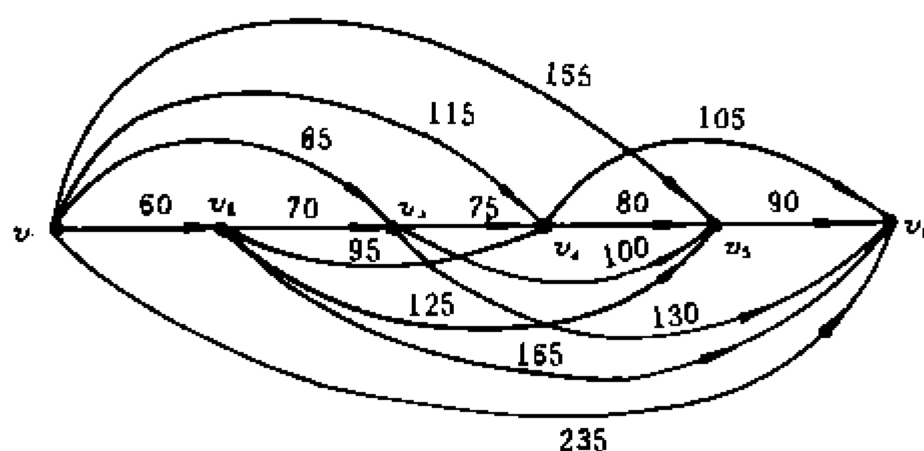


图 3-1-4

上图中每一条从 v_1 到 v_6 的路径都表达了一种设备更新的策略。例如(v_1,v_6)边便是第一年购置设备，一直用到最后一年的策略，总费用为 215 单位，$v_1\to v_2\to v_3\to v_4\to v_5\to v_6$ 又代表另一种极端情形，即当年买的设备，第二年便更新，总费用为

$$60+70+75+80+90=375 \text{ 单位。}$$

用戴克斯徒拉(Dijkstra)法求 v_1 到 v_6 的最佳路径如下图 3-1-5，从而可得最短路径为 $v_1\to v_4\to v_6$，即第一年购置，到第四年更新一次，直到最后为最佳方案，总费用为 220 单位。

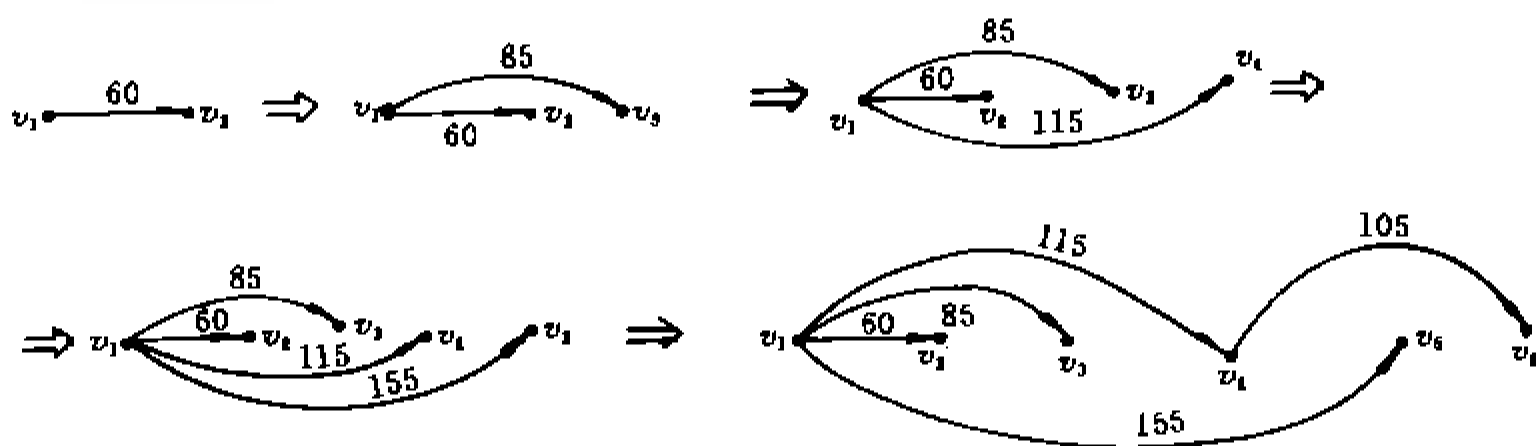


图 3-1-5

§2 最短树问题及其算法

具有 n 个顶点的树中，某树的树枝上的权之和称为该树的权。具有最小权的树称为最短树。首先我们来看一看具有 n 个顶点的树的数目是多少？

定理 过 n 个顶点的树的数目为 n^{n-2}。

证明 假设这 n 个顶点的标号为 $1,2,\cdots,n$。T 是其中的一棵树，T 的树叶中标号最

小的设为a_1，其邻接点为b_1，移去a_1点和边(a_1,b_1)，在剩下的$n-1$个顶点的树的树叶中，寻找某标号最小的，设为a_2，其邻接点为b_2，移去顶点a_2和边(a_2,b_2)，…，继续以上的步骤，直到最后剩下一条边为止。这样共移走了$n-2$条边，可得相应的序列：

$$b_1b_2\cdots b_{n-2}。$$

也就是说按照上面的步骤，树T唯一地对应了一个具有$n-2$个数的序列：$b_1b_2\cdots b_{n-2}$，其中b_i可以相同，也可以不同。

反之，给定一个具有$n-2$个数的序列：

$$b_1b_2\cdots b_{n-2}。\tag{1}$$

可以找到一棵树与之对应。办法如下：

从序列

$$1,2,\cdots,n\tag{2}$$

中找出第一个不在序列(1)中出现的数，显然这个数就是a_1，建立边(a_1,b_1)，然后从序列(1)中消去b_1，从(2)中消去a_1，在余下的序列(1)和(2)中继续以上的过程，直到序列(1)空为止。这时在序列(2)中剩下两个数，设为a_i，b_i，连接点a_i和b_i便得一棵树。

这样，n个标号的顶点的树T和$n-2$个数$b_1,b_2,\cdots,b_{n-2}$组成的序列(1)建立了一一对应关系。而序列(1)的数目为n^{n-2}，从而可知树的数目也为n^{n-2}。

上面给出了n个顶点的树的数目为n^{n-2}，若用枚举法从n^{n-2}个树中找出最短树，将是极大的计算量，故有必要寻找更好的算法来确定最短树。

寻求最短树的问题也是有它实际背景的，例如在某一国家或地区，需要营造一铁路网把一些城市联结起来，要求总长度最短或造价最低。例如图 3-2-1 是 15 座城市及彼此之间的距离，求其最短树。

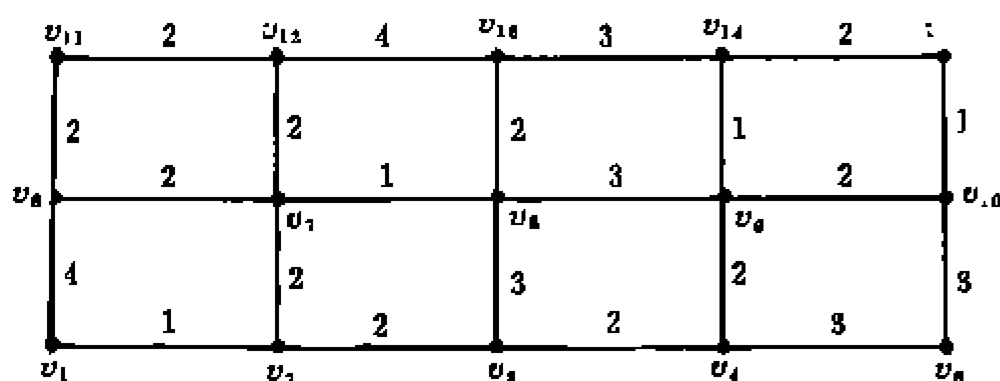


图 3-2-1

求最短树的步骤如下：

(1) 先任取一支撑树(见图 3-2-2(a))。

(2) 若加上一条余树的边，立即形成一个回路。如果这回路中有一条边比加上的边要长，则以所加的边取代较长的边，形成新的树。这样进行下去直到不出现这现象为止。

如图 3-2-2(b)—(f)便是这个修改更新的过程。

求最短树的 Kruskal 算法：

图G的顶点数为n，边数为m。

首先把赋权图G的边按权的递增顺序排列：

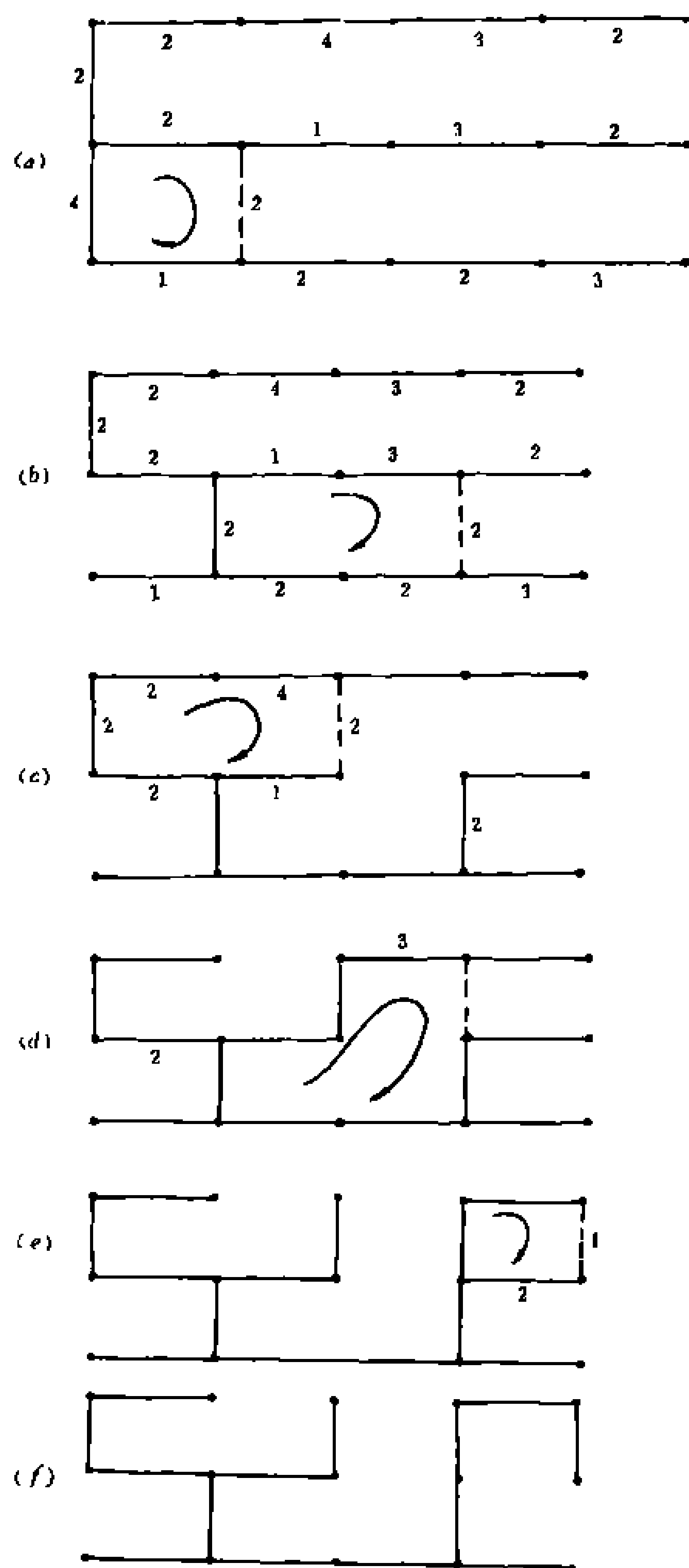


图 3-2-2

$$e_1 < e_2 < \cdots < e_m$$

先取进 e_1 边，即 $T\leftarrow\{e\}$。检查 e_2 边，如果 e 边和 e_2 边不构成回路，则取进 e_2 边，即作 $T\leftarrow T\cup\{e_2\}$；否则放弃 e_2 边。再检查 e_3 边，如若 e_3 边和 e_1，e_2 不构成回路，则作 $T\leftarrow T\cup\{e_3\}$，即将 e_3 边取作树 T 的树枝，否则将 e_3 放弃，如此反复继续下去，直到找到 $n-1$ 条边

加到树 T 为止。这便是所求的最短树。

Kruskal 算法的基本思想是在不构成回路的条件下"择优录取"权小的边。算法的有效性的证明略去。

上面的例子用 Kruskal 算法叙述如下：

（1）把图 3-2-3 的 22 条边按权从小到大顺序排列：

$d_{12}^{(1)}=1$， $d_{78}^{(2)}=1$， $d_{9,14}^{(3)}=1$，
$d_{10,15}^{(4)}=1$， $d_{23}^{(5)}=2$， $d_{34}^{(6)}=2$，
$d_{27}^{(7)}=2$， $d_{49}^{(8)}=2$， $d_{67}^{(9)}=2$，
$d_{9,10}^{(10)}=2$， $d_{6,11}^{(11)}=2$， $d_{7,12}^{(12)}=2$，
$d_{8,13}^{(13)}=2$， $d_{11,12}^{(14)}=2$， $d_{14,15}^{(15)}=2$，
$d_{45}^{(16)}=3$， $d_{3,8}^{(17)}=3$， $d_{5,10}^{(18)}=3$，
$d_{8,9}^{(19)}=3$， $d_{13,14}^{(20)}=3$， $d_{1,6}^{(21)}=4$，
$d_{12,13}^{(22)}=4$。

（2）把上面的各边顺序逐个加上去，若出现回路则把这条边排除。图 3-2-3 中各边的括号（ ）里的数是该边的排序。

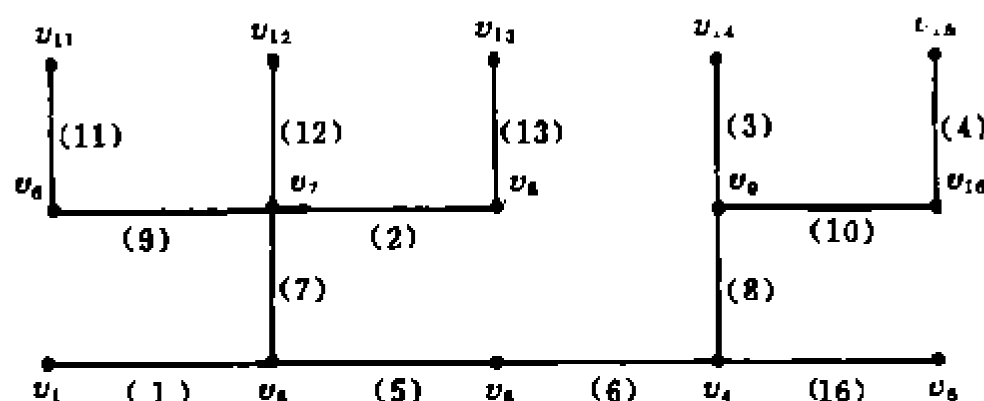


图 3-2-3

算法步骤：

（1）将边集合 E 的边依权的大小从小到大的顺序排列得

$$e_1, e_2, \cdots, e_m.$$

（2）$T\leftarrow\{e_1\}$，$i\leftarrow 1$，$j\leftarrow 2$。

（3）若 $i=n-1$ 则作【输出 T，结束.】，否则转（4）。

（4）若将 e_j 加到 T 上形成一回路则作【$j\leftarrow j+1$，转第 4 步.】，否则转（5）。

（5）$T\leftarrow T\cup\{e_j\}$，$j\leftarrow j+1$，$i\leftarrow i+1$，转（3）。

这里 i 和 j 分别用以记录加入到生成树的边的数目和依次进行检查的边的序号。【 】在这里是作为语句的括号，相当于 begin 和 end 的作用。

Kruskal 算法用框图形式表示见图 3-2-4。

和 Kruskal 算法齐名的还有 Prim 算法，Kruskal 算法要求将图的边按权的大小从小到大排列。Prim 算法则不需要，它的基本思想是：从某一点开始，设为 v_1，则作 $S\leftarrow\{v_1\}$。求 $V\backslash S$ 中的点与 S 中点间距离最短者。设为 (v_k, v_j)，其中 $v_k\in S$，$v_j\in V\backslash S$，则将 (v_k, v_j) 边收

入到树 T 中来，且 v_j 进入 S。依此反复进行，直到 n 个顶点用 $n-1$ 条边接起来为止。下面算法假定 $D=(d_{ij})_{n\times n}$ 是图 G 的距离矩阵，若 $(v_i, v_j)\notin E$ 时，$d_{ij}=\infty$，而且 $d_{ii}=\infty$。$i,j=1, 2, \cdots, n$。

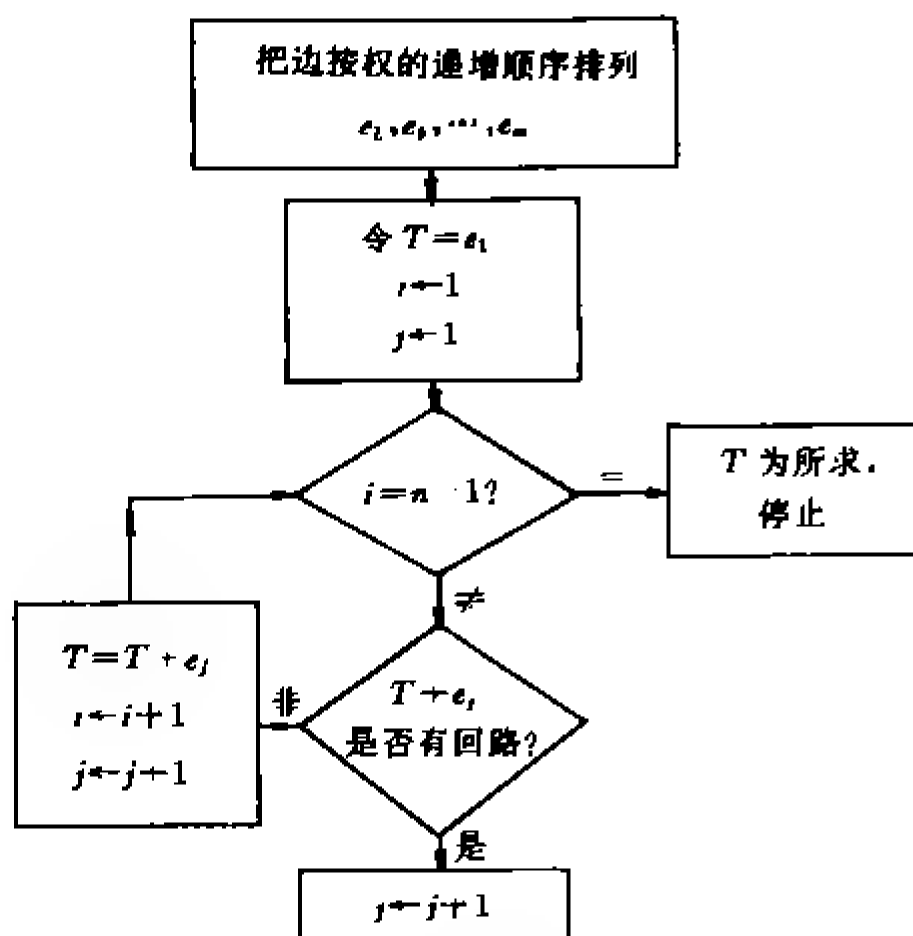


图 3.24

Prim 算法步骤：

(1) $T\leftarrow\Phi$, q【1】$\leftarrow 0$, $k\leftarrow 1$,

　　i 从 2 到 n 作

　　【 $q[i]\leftarrow d_{1i}$, $p[i]\leftarrow 1$,】。

(2) 若 $k\geq n$ 则作

　　【 输出 T，结束，】，否则作

　　【 min$\leftarrow\infty$,

　　　j 从 2 到 n 作

　　　【 若 $0<q[j]<$min 则作

　　　　【 min$\leftarrow q[j]$, $h\leftarrow j$】】】。

(3) $T\leftarrow T\cup\{(h, p[h])\}$, $q[h]\leftarrow 0$。

(4) j 从 2 到 n 作

　　【 若 $d_{hj}<q[j]$ 则作

　　　【 $q[j]\leftarrow d_{hj}$, $p[j]\leftarrow h$】】。

(5) $k\leftarrow k+1$，转(2)。

算法中数组 $p[i]$ 用以记录和 v_i 点最接近属于 S 中的点，$q[i]$ 则用以记录它们的距离。$q[i]=0$ 表示 v_i 已收入 S。

图 3-2-5 是用 Prim 算法求最短树的例。

例：求图 3-2-5(a)的最短树。

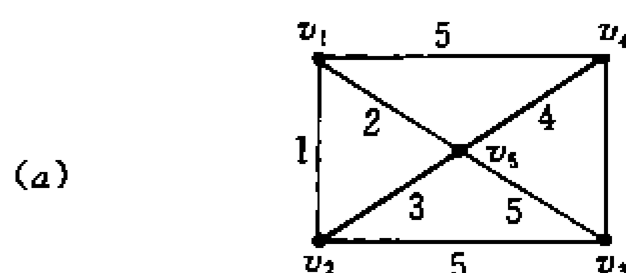


图 3-2-5(a)

解：以 v_3 为起点。

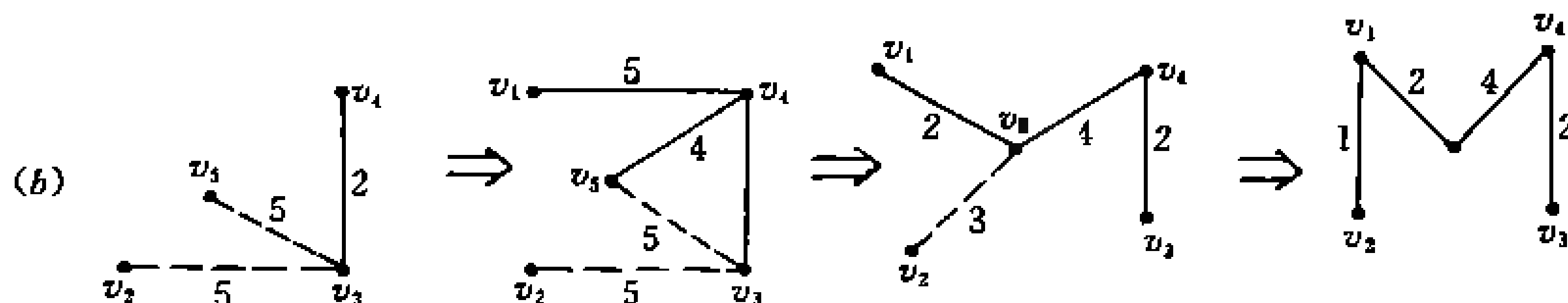


图 3-2-5(b)

§3 任意两点间最短距离及其算法

本章第一节给出了求图中给定点到其它各点的最短路径。现在讨论任意两点间的最短距离。当然利用给定点到其它各点最短距离的算法也可以求任意两点间的最短距离。不过后者有自己的特殊性，读者将可看到有些算法是十分聪明的。

为了讨论方便，先定义矩阵的一种运算。

设 $A=(a_{ij})_{m\times l}$，$B=(b_{jk})_{l\times n}$，

定义 $C=A \circledast B=(c_{ij})_{m\times n}$，

其中 $c_{ij}=\min\{a_{i1}+b_{1j},\ a_{i2}+b_{2j},\cdots,a_{il}+b_{lj}\}$。

定义 设$A=(a_{ij})_{m\times n}$，$B=(b_{ij})_{m\times n}$

$$D=(d_{ij})_{m\times n}=A \oplus B,$$

其中 $d_{ij}=\min\{a_{ij},\ b_{ij}\}$。

可以利用上面定义的运算求任意两点间的最短距离。设 $D=(d_{ij})_{n\times n}$是图 G 的距离矩阵，即 d_{ij} 表示 v_iv_j 边的长度。但若 v_i 与 v_j 无边相连时，则 $d_{ij}=\infty$。这样

$$D^{(2)}=D \circledast D=(d_{ij}^{(2)})_{n\times n}$$

其中 $d_{ij}^{(2)}=\min\{d_{i1}+d_{1j},\ d_{i2}+d_{2j},\cdots,d_{in}+d_{nj}\}$。

故 $d_{ij}^{(2)}$ 表示从 v_i 到 v_j 两步可到的道路中的距离最短者。同理 $D^{(k)}=D^{(k-1)} \circledast D=(d_{ij}^{(k)})_{n\times n}$的意义是，$d_{ij}^{(k)}$ 为从 v_i 出发 k 步到 v_j 的道路中距离最短者。

于是矩阵

$$S = D \oplus D^{(2)} \oplus \cdots\cdots \oplus D^{(n)} = (s_{ij})_{n\times n}$$

的意义是：s_{ij}为 v_i 到 v_j 的所有道路中的距离最短者。

例：

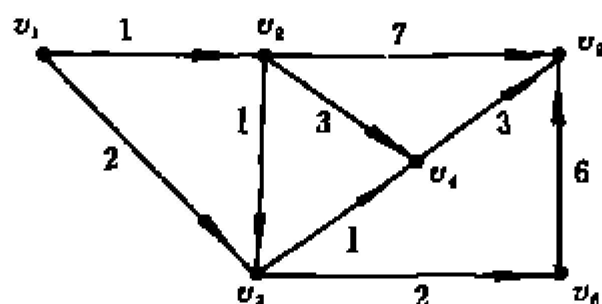


图　3 3-1

$$D=\begin{array}{c} v_1\\ v_2\\ v_3\\ v_4\\ v_5\\ v_6 \end{array}\begin{bmatrix} \infty & 1 & 2 & \infty & \infty & \infty\\ \infty & \infty & 1 & 3 & \infty & 7\\ \infty & \infty & \infty & 1 & 2 & \infty\\ \infty & \infty & \infty & \infty & \infty & 3\\ \infty & \infty & \infty & \infty & \infty & 6\\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix},$$
$$\qquad\; v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \quad v_6$$

$$D^{(2)}=\begin{bmatrix} \infty & 1 & 2 & \infty & \infty & \infty\\ \infty & \infty & 1 & 3 & \infty & 7\\ \infty & \infty & \infty & 1 & 2 & \infty\\ \infty & \infty & \infty & \infty & \infty & 3\\ \infty & \infty & \infty & \infty & \infty & 6\\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix} * \begin{bmatrix} \infty & 1 & 2 & \infty & \infty & \infty\\ \infty & \infty & 1 & 3 & \infty & 7\\ \infty & \infty & \infty & 1 & 2 & \infty\\ \infty & \infty & \infty & \infty & \infty & 3\\ \infty & \infty & \infty & \infty & \infty & 6\\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}$$

$$=\begin{bmatrix} \infty & \infty & 2 & 3 & 4 & 8\\ \infty & \infty & \infty & 2 & 3 & 6\\ \infty & \infty & \infty & \infty & \infty & 4\\ \infty & \infty & \infty & \infty & \infty & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}。$$

类似可得

$$D^{(3)}=\begin{bmatrix} \infty & \infty & \infty & 3 & 4 & 6\\ \infty & \infty & \infty & \infty & \infty & 5\\ \infty & \infty & \infty & \infty & \infty & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty\\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix},$$

$$
D^{(4)} = \begin{bmatrix} \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix},
$$

$$
D^{(5)} = (d_{ij}^{(5)})_{6\times 6},\ d_{ij}^{(5)} = \infty,\ i,\ j = 1,\ 2,\ \cdots,\ 6。
$$

$\therefore\ S = D \oplus D^{(2)} \oplus D^{(3)} \oplus D^{(4)} = (s_{ij})$，

$$
S = \begin{bmatrix} \infty & 1 & 2 & 3 & 4 & 6 \\ \infty & \infty & 1 & 2 & 3 & 5 \\ \infty & \infty & \infty & 1 & 2 & 4 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix}
$$

上面矩阵中若 v_i，v_j 两个顶点之间没有道路时，则出现 $s_{ij}=\infty$，在机器中实际是一个标志数，凡遇到这个数时便按∞处理，这个数可以是机器所允许的充分大的一个数。

可以看出以上所介绍的计算任意两点间最短距离的算法的计算量是 n^4，下面介绍一种求任意两点间最短距离的 Warshall 算法。

设图 G 的顶点数为 n，D 是其距离矩阵，Warshall 算法的步骤如下：

(1) 输入 D；

(2) $k \leftarrow 1$；

(3) $i \leftarrow 1$；

(4) $d_{ij} = \min(d_{ij},\ d_{ik}+d_{kj})$，$j=1,\ 2,\ \cdots,\ n$；

(5) $i \leftarrow i+1$，若 $i \leqslant n$ 转(4)；

(6) $k \leftarrow k+1$，若 $k \leqslant n$ 转(3)，否则停止。

Warshall 算法的框图如图 3-3-2 所示。

算法是对 i,j,k 进行循环，故它的复杂性是 $O(n^3)$，即对矩阵 D 进行 k 次修改。还是以图 3-3-2 为例：

$$
D^{(1)} = \begin{bmatrix} \infty & 1 & 2 & \infty & \infty & \infty \\ \infty & \infty & 1 & 3 & \infty & 7 \\ \infty & \infty & \infty & 1 & 2 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & \infty & \infty & \infty & \infty & 6 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix},
$$

$k=2$ 时，$d_{ij} \leftarrow \min(d_{ij},\ d_{i2}+d_{2j})$，得

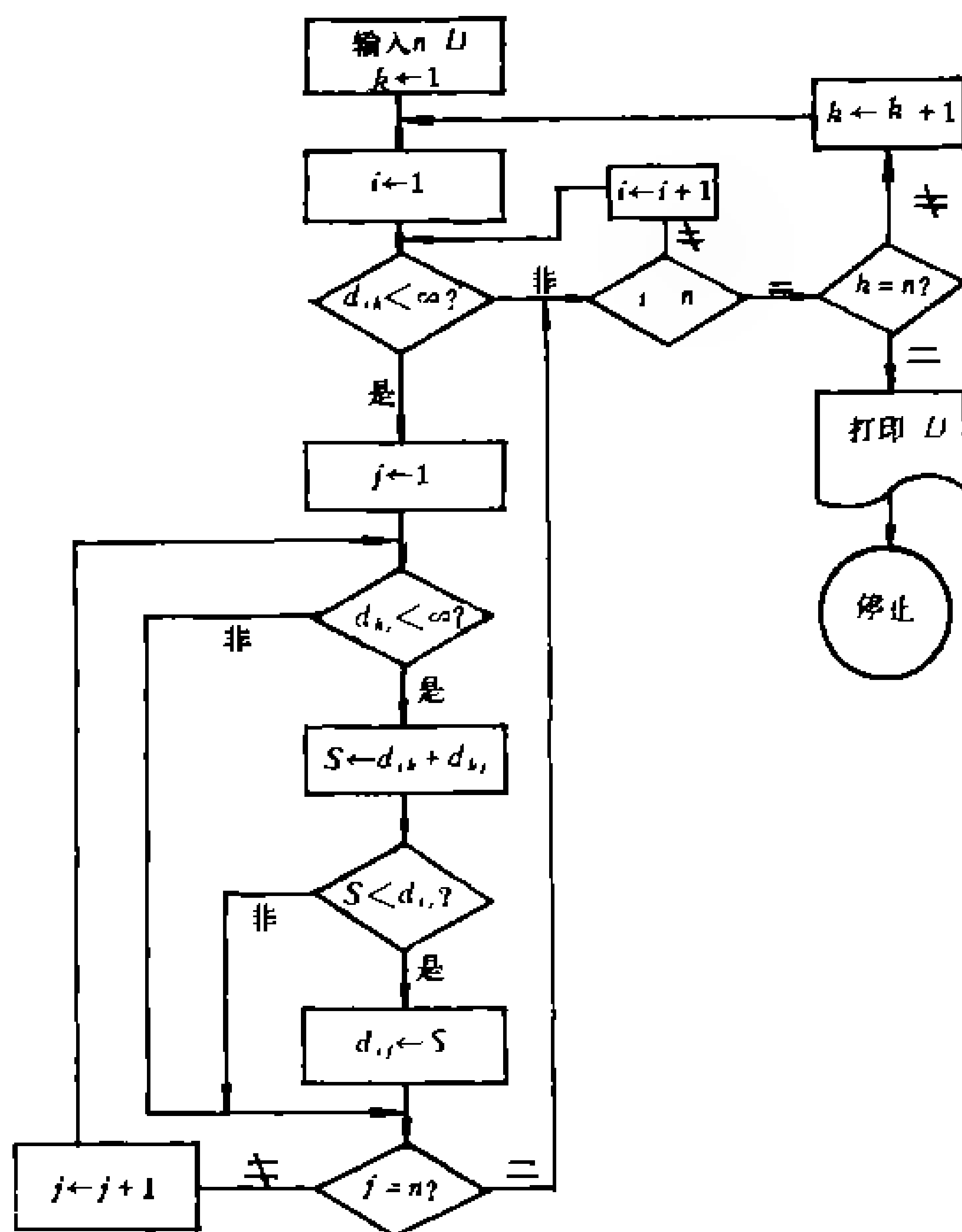


图 3-3-2

$$D^{(2)}=\begin{bmatrix}\infty & 1 & 2 & 4^{*} & \infty & \infty\\ \infty & \infty & 1 & 3 & \infty & 7\\ \infty & \infty & \infty & 1 & 2 & \infty\\ \infty & \infty & \infty & \infty & \infty & 3\\ \infty & \infty & \infty & \infty & \infty & 6\\ \infty & \infty & \infty & \infty & \infty & \infty\end{bmatrix}。$$

* 是新修改元素的符号，例如 $d_{14}=d_{12}+d_{24}=4<\infty$，$k=3$ 时有

$$D^{(3)}=\begin{bmatrix}\infty & 1 & 2 & 3^{*} & 4^{*} & \infty\\ \infty & \infty & 1 & 2^{*} & \infty & \infty\\ \infty & \infty & \infty & 1 & 2 & 7\\ \infty & \infty & \infty & \infty & \infty & 3\\ \infty & \infty & \infty & \infty & \infty & 6\\ \infty & \infty & \infty & \infty & \infty & \infty\end{bmatrix}。$$

其中 $d_{14}=d_{13}+d_{34}=2+1=3<4$,

$d_{15}=d_{13}+d_{35}=4$。

$d_{24}=d_{23}+d_{34}=1+1=2<3$,

$k=4,5,6$ 依此类推。

§4 图的连通性判断

设 $G=(V,E)$ 为无向图,其中

$$V=\{v_1, v_2, \cdots, v_n\},\ n=|V|,\ E=\{e_1, e_2, \cdots, e_m\},\ m=|E|。$$

令

$$e_i=(v_{a_i}, v_{b_i}),\ i=1, 2, \cdots, m。$$

连通图的概念前面已讨论过,现在的问题是判断图 G 是否连通?若不是连通的,有几个连通块?

下面算法的基本步骤是程式化了的。

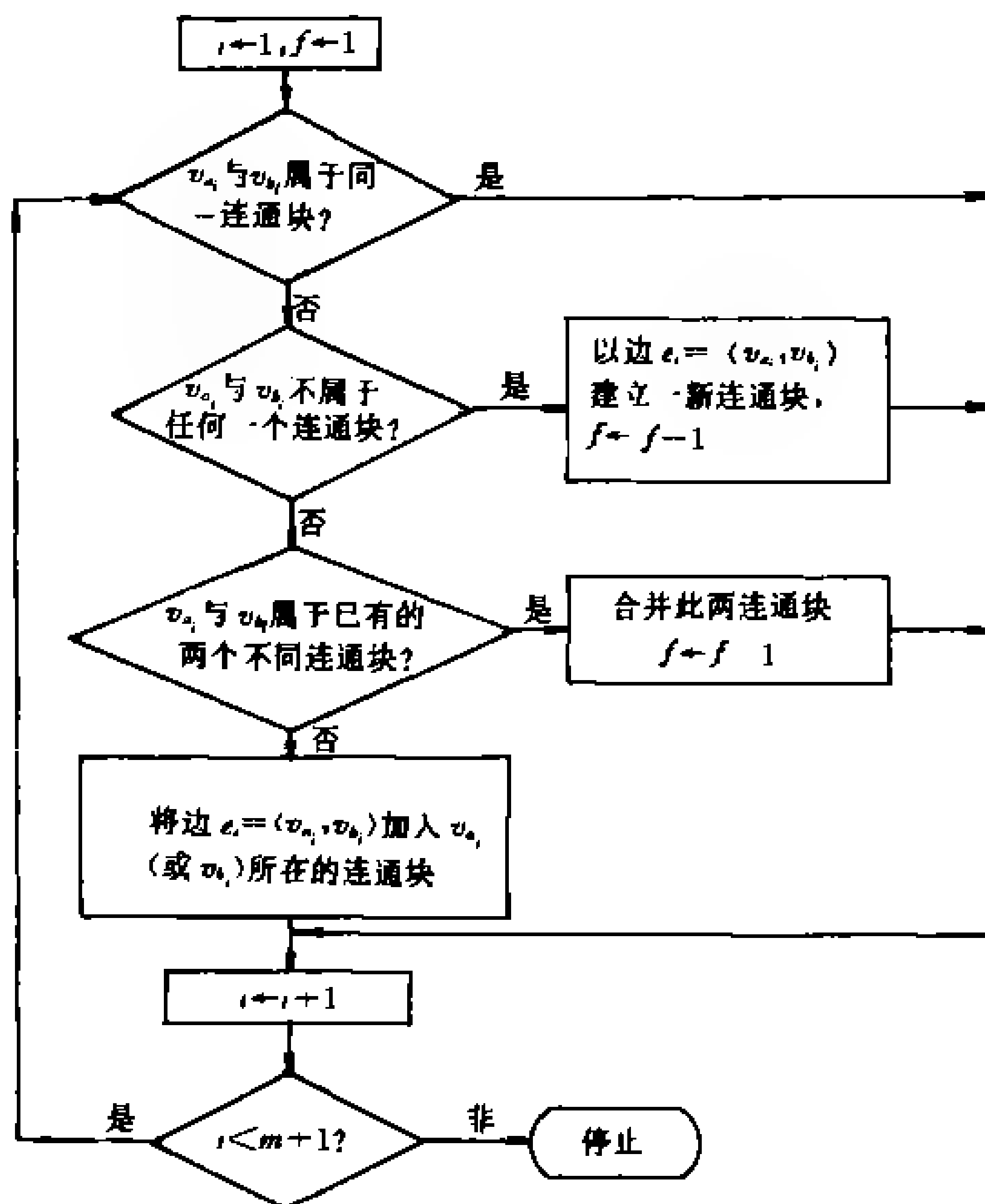


图 3-4-1

(1) $i\leftarrow 1$, $f\leftarrow 1$。

(2) 若 v_{a_i} 与 v_{b_i} 同属于一个连通块时转(6);否则转(3)。

(3) 若 v_{a_i} 与 v_{b_i} 都不属于任何一个连通块，则用边 $e_i=(v_{a_i},v_{b_i})$ 建立一新的连通块，$f\leftarrow f+1$，转(6)。

(4) 若 v_{a_i}(或 v_{b_i})属于第 p 个连通块，但 v_{b_i}(或 v_{a_i})不属于任何一个连通块，则边 $e_i=<v_{a_i}, v_{b_i}>$ 也属于第 p 个连通块；转(6)。

(5) 若 v_{a_i} 与 v_{b_i} 分别属于不同的两个连通块，则边 $e_i=(v_{a_i},v_{b_i})$ 使这两个不同的连通块连成一个连通块，$f\leftarrow f-1$，转(6)。

(6) $i\leftarrow i+1$，若 $i<m+1$，转(2)；否则结束。

上面算法可用框图描述，见图 3-4 1。

§5 树的生成

上节讲的判断连通块的算法，略加修改便可以用来生成一棵树，只要当 $e_i=(v_{a_i}, v_{b_i})$ 边的两端点已给了所属的连通块标志时，e_i 不作任何处理即可。细节问题留给读者自己思考。

生成图 G 的所有的树，则比较复杂。下面首先介绍几种算法：

1. 在第二章我们介绍了从 Binet Cauchy 定理导出树的计数公式，即连通图 G 的树的数目为

$$\det(\boldsymbol{B}_k\boldsymbol{B}_k^T),$$

其中 B_k 为 G 的基本关联矩阵，类似的办法可以用来给出树的“清单”。为此引进与矩阵 $\boldsymbol{B}_k=(b_{ij})$ 对应的矩阵 $\boldsymbol{B}_k'=(b_{ij}')$，其中

$$b_{ij}'=\begin{cases} e_j, & b_{ij}=1; \\ -e_j, & b_{ij}=-1; \\ 0, & b_{ij}=0. \end{cases}$$

或简记为

$$b_{ij}'=b_{ij}\cdot e_j,$$

则

$$\det(\boldsymbol{B}_k'\boldsymbol{B}_k^T)$$

给出树的清单。这个行列式的展开式中不同的项代表不同的树。这个公式的证明和第二章第 7 节几乎完全一样，不重述。例：

$$\boldsymbol{B}_k=\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \end{pmatrix},$$

$$\therefore \qquad \boldsymbol{B}_k'=\begin{pmatrix} e & e_2 & e_3 & 0 & 0 & 0 \\ -e_1 & 0 & 0 & e_4 & e_5 & 0 \\ 0 & -e_2 & 0 & e_4 & 0 & e_6 \end{pmatrix}.$$

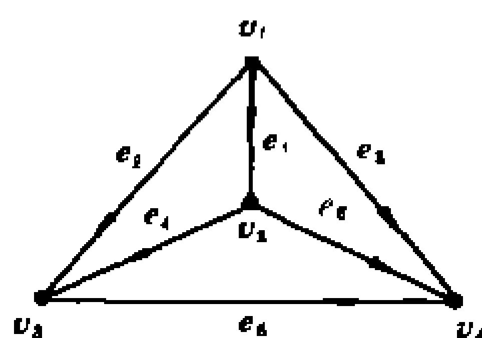


图 3-5-1

$$B'_4B_4^{\mathrm{T}}=\begin{pmatrix} e_1 & e_2 & e_3 & 0 & 0 & 0\\ -e_1 & 0 & 0 & e_4 & e_5 & 0\\ 0 & -e_2 & 0 & -e_4 & 0 & e_6\end{pmatrix}\begin{pmatrix}1 & -1 & 0\\ 1 & 0 & -1\\ 1 & 0 & 0\\ 0 & 1 & 1\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}$$

$$=\begin{pmatrix} e_1+e_2+e_3 & -e_1 & -e_2\\ -e_1 & e_1+e_4+e_5 & -e_4\\ -e_2 & -e_4 & e_2+e_4+e_6\end{pmatrix},$$

$$\therefore\ \det(B'_4B_4^{\mathrm{T}})=\begin{vmatrix} e_1+e_2+e_3 & -e_1 & -e_2\\ -e_1 & e_1+e_4+e_5 & -e_4\\ -e_2 & -e_4 & e_2+e_4+e_6\end{vmatrix}$$

$$=(e_1+e_2+e_3)(e_1+e_4+e_6)(e_2+e_4+e_6)-e_1e_2e_4-e_1e_2e_4-e_2^2(e_1+e_4+e_5)$$
$$e_4^2(e_1+e_2+e_3)-e_1^2(e_2+e_4+e_5)$$
$$=e_1e_4e_6+e_1e_5e_2+e_1e_5e_4+e_1e_5e_6+e_1e_2e_6+e_2e_4e_6+e_2e_5e_4+e_2e_5e_6+e_1e_2e_3+e_1e_3e_4$$
$$+e_1e_3e_6+e_2e_3e_4+e_3e_4e_6+e_2e_3e_5+e_3e_4e_5+e_3e_5e_6。$$

2. 再引进一个叫做子图多项式的，即对于某一子图 G_i 定义：

$$P_i=a_1e_1+a_2e_2+\cdots+a_me_m=\sum_{i=1}^{m}a_ie_i,$$

其中系数

$$a_i=\begin{cases}1, & e_i\in E(G_i);\\ 0, & 其它。\end{cases}$$

见图 3-5-2。

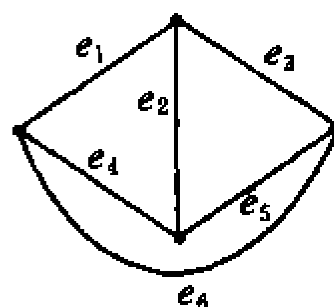


图 3-5-2

$T=\{e_1, e_2, e_3\}$ 是它的树，则

$$P_T=e_1+e_2+e_3。$$

规定运算规则：

$$e_i+e_i=0,\qquad e_i\cdot e_i=0。$$

设 $\quad P_a=\sum_{i=1}^{m}a_ie_i,\quad P_b=\sum_{j=1}^{m}b_je_j$。

则 $\quad P_aP_b=\sum_{i=1}^{m}\sum_{j=1}^{m}a_ib_je_ie_j$。

例：$P_a=e_1+e_2+e_3$，$P_b=e_1+e_4+e_5$。

由于 $e_1\cdot e_1=0,\qquad e_1+e_1=0,$

$P_a+P_b=e_2+e_3+e_4+e_5,$

$P_a\cdot P_b=(e_1+e_2+e_3)\cdot(e_1+e_4+e_5)$

$\qquad=e_1e_4+e_1e_5+e_{21}+e_{24}+e_{25}+e_{31}+e_{34}+e_{35}$。

定理 设 $\{P_{s_1},P_{s_2},\cdots,P_{s_{n-1}}\}$ 是图 G 的基本割集多项式，则

$$\prod_{i=1}^{n}P_{s_i}=P_{s_1}\cdot P_{s_2}\cdots P_{s_{n-1}}=P_{T_1}+P_{T_2}+\cdots+P_{T_N}。$$

其中 N 是图 G 的树的数目，

$$P_{T_j}=e_{j_1}e_{j_2}\cdots e_{j_{n-1}}$$

对应于树：$T_j=\{e_{j_1},e_{j_2},\cdots,e_{j_{n-1}}\}$。

在给出这个定理的证明之前，先看一个例子这个定理究竟说明的是什么意思，特别是搞清楚建立运算法则 $e_i+e_i=0$，$e_i\cdot e_i=0$ 的含意，进而证明就不难。

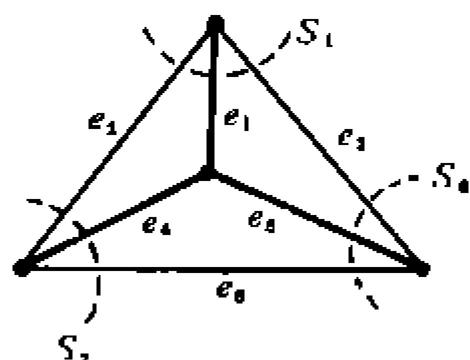


图 353

例：上图对应于树 $T=\{e_1,e_4,e_5\}$ 的基本割集多项式为：

$P_{s_1}=e_1+e_2+e_3,$

$P_{s_2}=e_2+e_4+e_6,$

$P_{s_3}=e_3+e_5+e_6,$

$P_{s_1}P_{s_2}P_{s_3}=(e_1+e_2+e_3)(e_2+e_4+e_6)(e_3+e_5+e_6)$

$\qquad=(e_1e_2+e_1e_4+e_1e_6+e_{24}+e_{26}+e_{32}+e_{34}+e_{36})(e_3+e_5+e_6)$

$\qquad=e_1e_2e_3+e_1e_2e_5+e_1e_2e_6+e_1e_3e_4+e_1e_4e_5+e_1e_4e_6+e_1e_3e_6+e_1e_5e_6$

$\qquad\quad+e_2e_3e_4+e_2e_4e_5+e_2e_4e_6+e_2e_5e_6+e_2e_3e_5+e_3e_4e_5+e_3e_4e_5+e_3e_5e_6$。

展开式右端共16项，其中每一项正好对应一棵树。从这例子可见：图 G 的所有的树可以看作是由树 $\{e_1,e_4,e_5\}$ 形成的，形成的办法是树枝 e_1 由对应的基本割集 $S_1=\{e_1,e_2,e_3\}$ 中一条边所取代，树枝 e_4 由对应的割集 $S_2=\{e_2,e_4,e_6\}$ 中的一条边所取代，e_5 由对应的割集 $S_3=\{e_3,e_5,e_6\}$ 中的一条边所取代而成的，然而 e_2 边同时存在于 S_1 和 S_2 中，e_3 边同时存在于 S_1 和 S_3 中，e_6 同时存在于 S_2 和 S_3 中，$e_1e_1e_3$ 项由于边数少于3，故不可能形成一棵树。又 $e_2e_3e_4$ 项出现了两次，它对应于一回路，故建立 $e_i\cdot e_i=0$，$e_i+e_i=0$ 的代数系统。

定理的证明：对于任意一树，$T_j=\{e_{j_1},e_{j_2},\cdots,e_{j_{n-1}}\}$，由于 S_1、S_2、$\cdots S_{n-1}$ 是基本割集，故有

$$\prod_{i=1}^{n-1}P_{s_i}=(\cdots+e_{j_1})(\cdots+e_{j_2})\cdots(\cdots+e_{j_{n-1}}),$$

即在 $\prod_{i=1}^{n-1} P_{i}$ 中对应一项 $e_{i_1}e_{i_2}\cdots e_{i_{n-1}}$。反过来，在展开式中除回路外的项都对应一棵树，而回路所对应的项必然出现两次，根据 $e_i+e_i=0$ 故消失了。

前面介绍的两个算法，从例子中可看出，计算过程中重复出现最后消失的项太多，这儿我们再介绍两个比较有效的算法。

3. Minty 算法

Minty 算法的思想是这样的：对于图 G 的任意一条特定的边 e，图 G 的树可以分成两大类：一类是含有 e 边的，另一类是不含 e 边的。

为此，令 G 图的 e 边短路，即让 e 边的两端点重合，得一新图 G_1，即 $G_1=G(\dot{e})$。又令图 G 中 e 边断路，即让 G 中的 e 边消去，得另一新图 G_2，即 $G_2=G(\bar{e})$，于是 G 图的树 $T(G)$ 由下面两部分组成：

1) G_1 图的树补上 e 边；

2) G_2 图的树。

或写成：

$$T(G)=\{T(G(\dot{e}))+\{e\}\}\cup\{T(G(\bar{e}))\}=\{T(G_1)+\{e\}\}\cup\{T(G_2)\}。$$

上式中 $T(G(\dot{e}))+\{e\}$ 表示由 $G(\dot{e})$ 图的树的集合中每一棵树，分别加上 e 边所形成新的树。

Minty 算法可以递归计算。

但 Minty 算法首先要求 e 边不是自环，而且不是"桥"（即任何回路都不含有的边，也就是任何一棵树必有的边）。如果有自环 e，则自动令之短路；如果 e 是桥，则图 G 可以分解成以 e 作桥的两部分，而且

$$T(G)=T(G(\dot{e}))+\{e\}。$$

在介绍 Minty 算法之前，先说明公式

$$T(G)=\{T(G(\dot{e}))+\{e\}\}\cup\{T(G(\bar{e}))\}$$ 的意义。

例如：

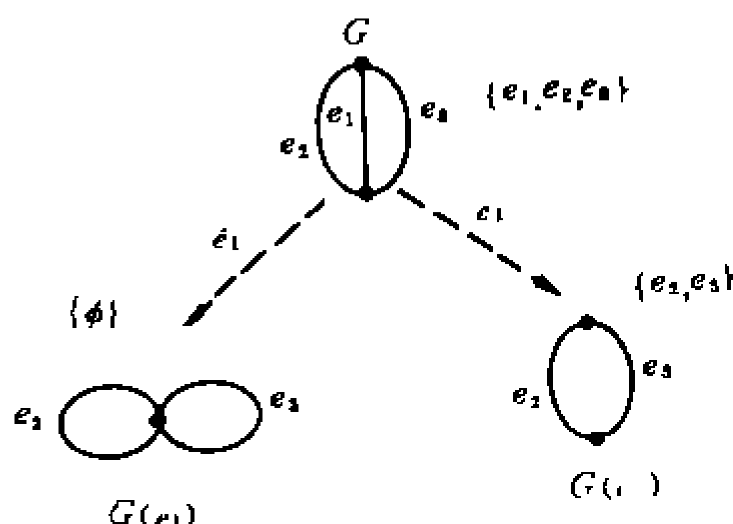


图 3-5-4

显然 $T(G(\dot{e}_1))=\varphi$，因为 $G(\dot{e}_1)$ 只含有两个自身回路，故不存在树。但 $G(\bar{e}_1)$ 可继续分解，如图 3-5-5。其中 $\dot{e}_2$ 表示 e_2 边短路，$\bar{e}_2$ 表示 e_2 边断路。

图旁边的{ }里的元素为与之相应的树的“清单”，例如，图 $G(\dot{e}_1\bar{e}_2)$ 的树为 $\{e_3\}$，显然图的“分解”是从上而下；然而树的形成则是自下而上的。

$$\therefore\ T(G(\bar{e}_1\dot{e}_2))=\Phi,$$

$$T(G(\dot{e}_1\dot{e}_2))=\{e_3\},$$

$$\therefore\ T(G(\bar{e}_1))=T(G(\bar{e}_1\dot{e}_2)+\{e_2\})\cup T(G(\bar{e}_1\bar{e}_2))$$
$$=\{e_2\}\cup\{e_3\}$$
$$=\{e_2,e_3\}。$$

但
$$T(G)=T(G(\dot{e}_1)+\{e_1\})\cup\{T(G(\bar{e}_1))\}$$
$$=\{e_1\}\cup\{e_2,e_3\}$$
$$=\{e_1,e_2,e_3\}。$$

图 3-5-5

所以图 3-5-6 的树有 3 棵，即图 3-5-7。

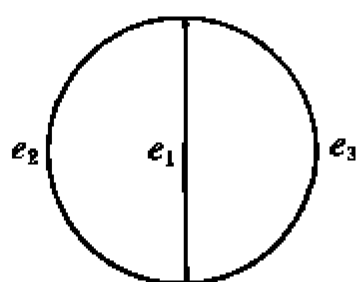


图 3-5-6

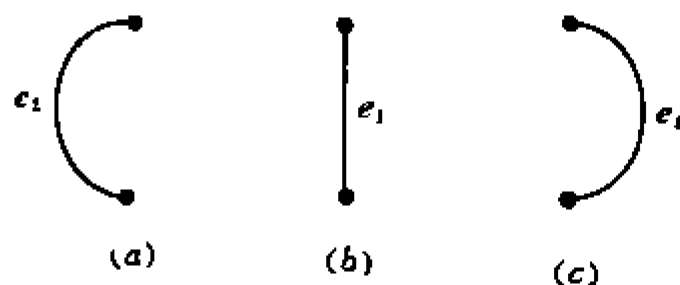


图 3-5-7

现举一例说明 Minty 算法(见图 3-5-8)。

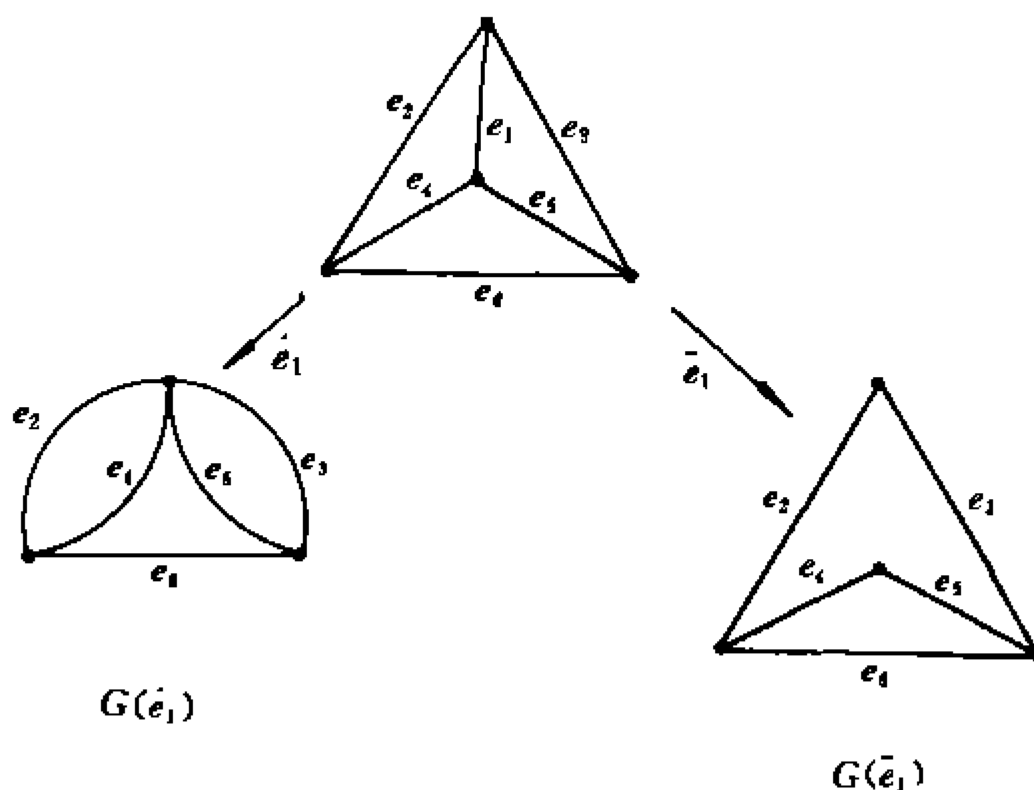


图 3-5-8

但

根据公式：

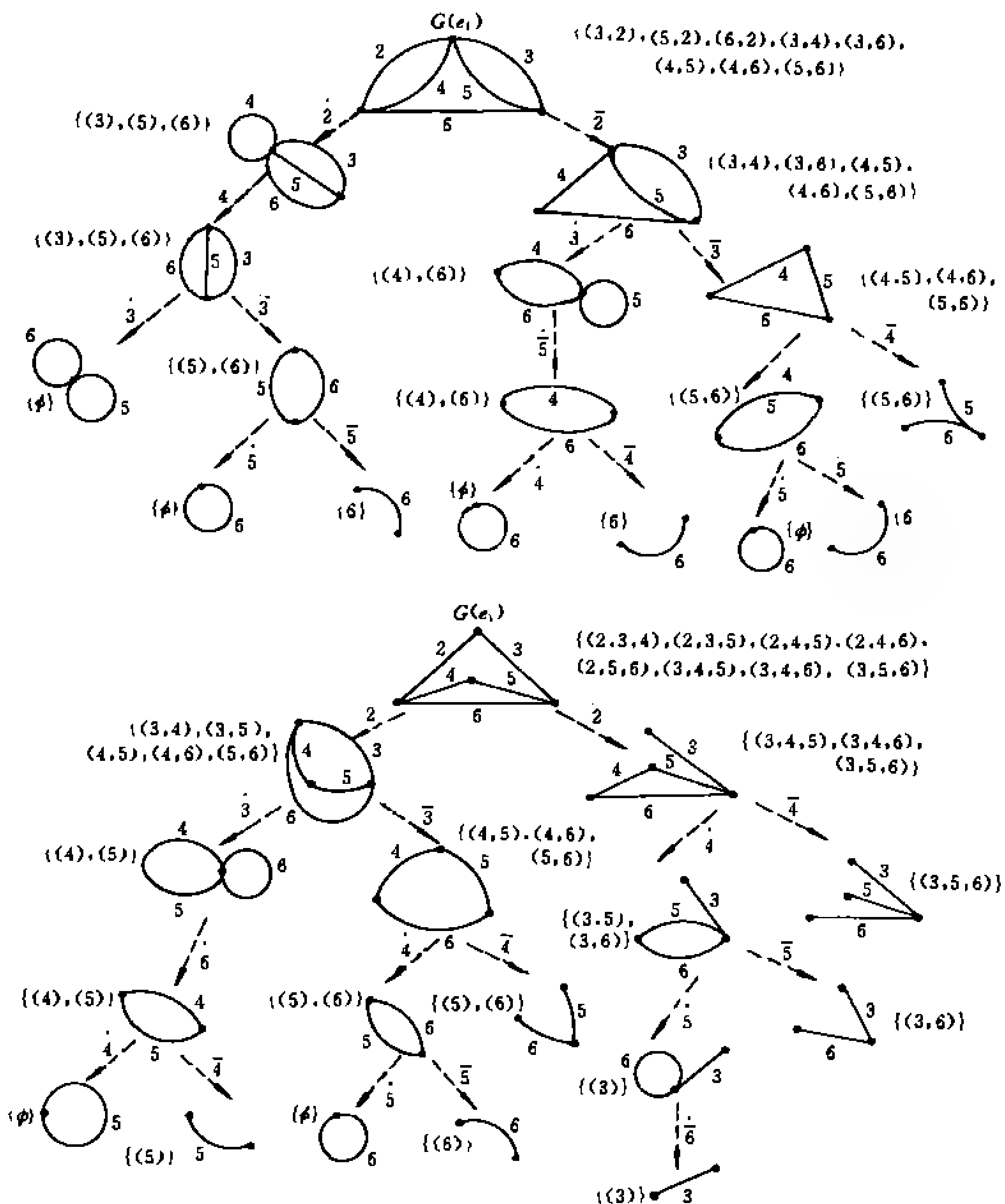


图 3-5-9

$$T(G) = \{T(G(e_1)) + \{e_1\}\} \cup T(G(\bar{e}_1))。$$

从下往上回溯，可得树 T 的"清单"如下：

(1,2,3)、(1,2,5)、(1,2,6)、(1,3,4)、(1,3,6)、(1,4,5)、(1,4,6)、(1,5,6)，

(2,3,5)、(2,3,4)、(2,4,5)、(2,4,6)、(2,5,6)、(3,4,5)、(3,4,6)、(3,5,6).

上面为方便起见，e_i 边就写为 i，比如(1,2,3)表示由 e_1,e_2,e_3 组成的一棵树。

4. Mayeda-Seshu 算法

它的基本思想是对于图 G 的一棵树 t_0，t_0 的每根树枝 e，对应一割集 $S_e(t_0)$；用 $S_e(t_0)$ 中的每一根余树枝取代 e 边，这样可得一组互不相同的树，即

$$T^{e_i} = \{t \mid t = t_0 \oplus \{e_i, e_i'\},\ e_i' \in S_{e_i}(t_0),\ e_i' \neq e_i\}。 \tag{1}$$

这里所得的树和 t_0 只有一条边不相同，而其它 $n-2$ 条边同属于 t_0。若考虑到与 t_0 有更多的不同的边，情况要复杂一些，为此引进

$$T^{e_{i_1}e_{i_2}\cdots e_{i_k}} = \{t \mid t = t_1 \oplus \{e_{i_k}, e_i'\},\ t_1 \in T^{e_{i_1}e_{i_2}\cdots e_{i_{k-1}}},\ e_i' \in S_{e_{i_k}}(t_1) \cap S_{e_{i_k}}(t_0),\ e_{i_k} \neq e_i\}。 \tag{2}$$

公式(2)是可以递归的，当 $k=1$ 时，便和(1)一致了。其中 $i_1 i_2 \cdots i_k$ 是自然序列 $1,2,\cdots,n-1$ 的一个子序列。设 $e_1, e_2, \cdots, e_{n-1}$ 为树 t_0 的树枝。

以图 3-5-10 为例介绍算法如下：

设取 $t_0 = \{e_1, e_2, e_3\}$，

$S_{e_1}(t_0) = \{e_1 e_4 e_6\}$，

$S_{e_2}(t_0) = \{e_2 e_4 e_5\}$，

$S_{e_3}(t_0) = \{e_3 e_5 e_6\}$。

$T^{e_1} = \{t \mid t = \{e_1 e_2 e_3\} \oplus \{e_1, e_i'\}, e_i' \in S_{e_1}(t_0), e_i' \neq e_1\}$

$= \{e_2 e_3 e_4,\ e_2 e_3 e_6\}$。

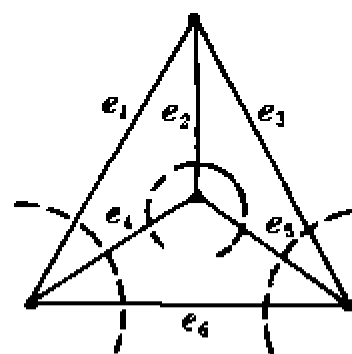


图 3-5-10

即 T^{e_1} 的特点是和树 t_0 相比，e_1 边被用其它的边所取代，其它两树枝 e_2, e_3 依然保存在树的树枝集合中。同理可得：

$$T^{e_2} = \{t \mid t = \{e_1 e_2 e_3\} \oplus \{e_2, e_i'\},\ e_i' \in S_{e_2}(t_0),\ e_i' \neq e_2\}$$

$$= \{e_1 e_3 e_4,\ e_1 e_3 e_5\};$$

$$T^{e_3} = \{e_1 e_2 e_5,\ e_1 e_2 e_6\}$$

根据定义

$$T^{e_1 e_2} = \{t \mid t = t_1 \oplus \{e_2, e_i'\},\ t_1 \in T^{e_1},\ e_i' \in S_{e_2}(t_1) \cap S_{e_2}(t_0),\ e_2 \neq e_i'\}$$

其中 $T^{e_1} = \{e_2 e_3 e_4,\ e_2 e_3 e_5\}$，

而　$S_{e_2}(e_2 e_3 e_4) = \{e_1, e_2, e_5, e_6\}$，

∴　$S_{e_2}(e_2 e_3 e_4) \cap S_{e_2}(t_0) = \{e_1, e_2, e_5, e_6\} \cap \{e_2, e_4, e_5\} = \{e_2, e_5\}$。

又　$S_{e_2}(e_2 e_3 e_6) = \{e_2, e_4, e_5\}$

∴　$S_{e_2}(e_2 e_3 e_6) \cap S_{e_2}(t_0) = \{e_2, e_4, e_5\} \cap \{e_2, e_4, e_5\} = \{e_2, e_4, e_5\}$，

∴　$T^{e_1 e_2} = \{e_3 e_4 e_5,\ e_3 e_4 e_5,\ e_3 e_5 e_6\}$。

即 $T^{e_1 e_2}$ 与 t_0 树相比较，只有一条树枝 e_3 相同，是不含树枝 e_1, e_2 的树的集合。

同理可得

$$T^{e_1 e_3} = \{e_2 e_4 e_5, e_2 e_4 e_6, e_2 e_5 e_6\};$$

$$T^{e_2 e_3} = \{e_1 e_4 e_6,\ e_1 e_4 e_5,\ e_1 e_5 e_6\};$$

$$T^{e_1 e_2 e_3} = \Phi。$$

这样共得树如下：

$$t_0 = e_1e_2e_3,$$
$$T^{e_1} = \{e_2e_3e_4,\ e_2e_3e_6\},$$
$$T^{e_2} = \{e_1e_3e_4,\ e_1e_3e_5\},$$
$$T^{e_3} = \{e_1e_2e_5,\ e_1e_2e_6\},$$
$$T^{e_1e_2} = \{e_3e_4e_5,\ e_3e_4e_5, e_3e_5e_6\},$$
$$T^{e_1e_3} = \{e_2e_4e_5,\ e_2e_4e_6, e_2e_5e_6\},$$
$$T^{e_2e_3} = \{e_1e_4e_5,\ e_1e_4e_6, e_1e_5e_6\}。$$

不难证明，由此产生的树都不相同，而且包含所有的树。证明从略。

§6 DFS 算 法

一、DFS 是英文“Depth First Search”的缩写，意为“深度优先搜索”，它是算法中最常用的一种

DFS 的基本思想可从下面 4×4 棋盘的布局问题得到形象地表达。4 个棋子布到棋盘上要求每行每列各一个，且不存在两个棋子在同一对角线上。4 个棋子中若第 1 个位于第 1 行的第 h 列；第 2 个棋子位于第 2 行的第 i 列；第 3、第 4 个棋子分别为第 3 行的第 j 列，和 4 行的第 k 列。则令这样的布局为 $hijk$。所以 4×4 棋盘的布局问题可用穷举法逐个搜索即 1,2,3,4 的每一排列对应一种布局。判断它是否满足要求的条件。以(1324)为例，当然这个布局的第 2 行和第 3 行的棋子在一对角线上，而且第 1 和第 4 行的棋子也是在同一对角线的，故不满足要求。这样 1,2,3,4 每一个排列对应一种布局。可采用穷举法进行搜索。即将 1,2,3,4 的全排列逐个进行审查，将符合要求的布局找出来。对于 4×4 的棋盘并不困难。全排列的全体仅 24 个。但推到 $n\times n$ 的棋盘问题，穷举法便出问题了。因 n 个元素的全排列有 $n!$ 种方案，根据 Stirling 公式

$$n! \approx \sqrt{2n\pi}\left(\frac{n}{e}\right)^n,$$

以 $n=26$ 为例

$$26! \approx 4 \times 10^{26},$$
$$365 \times 24 \times 3600 = 3.1536 \times 10^7。$$

即一年有 3.1536×10^7 秒，利用每秒能搜索 10^7 个排列的超高速电子计算机来处理 $n\times n$ 的棋盘问题，需要的时间。

$$T = \frac{4 \times 10^{26}}{3.1536 \times 10^{14}\text{年}} = 1.2684 \times 10^{-2}\text{年}。$$

可见穷举搜索作为一种方法理论上是可计算的，而实际上是不可能做到的。

对 4×4 的棋盘采用深度优先法可形象地用图 3-6-1 说明。图中× 表示不满足要求的格子。所以 DFS 搜索法可以形象地用一句话来概括：向前走碰壁回头。例如图 3 6 1 中(1)→(2)→(3)表示一种布子的过程，但在第 3 行“碰壁”。(4)表示“回头”后继续向前走，但在第 4 行“碰壁”，导致第 3 行，波及第 2 行都“碰壁”。(7)→(8)是“向前走”导致找到合格的布局(8)，对应于(2413)排列。

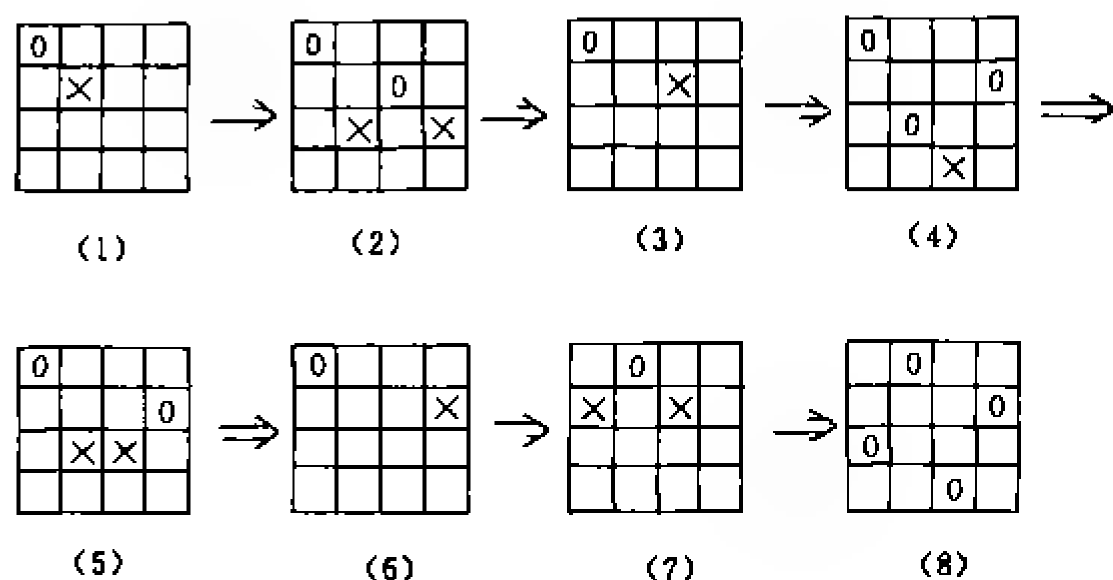


图 3-6-1

DFS 搜索法可以看作是在一搜索树(即搜索空间)上进行前进后退的过程，4×4 棋盘

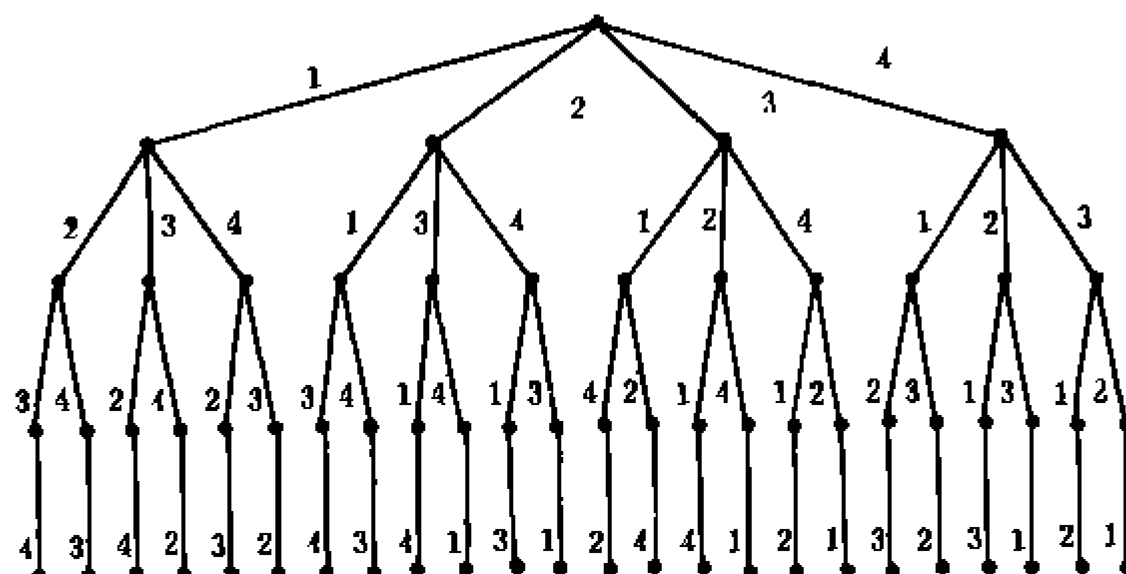


图 3-6-2

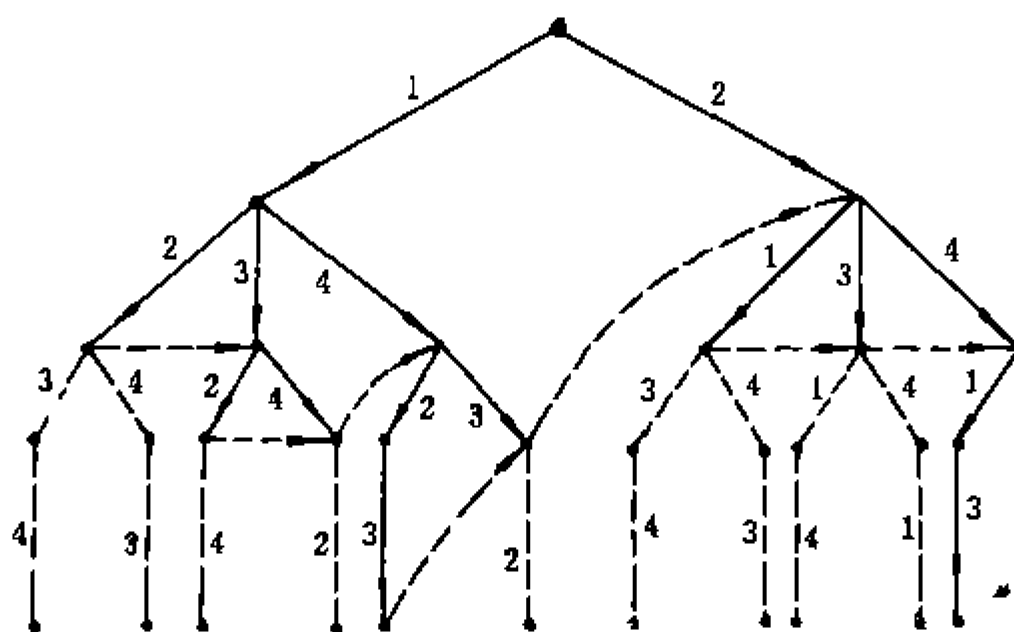


图 3 6 3

问题的搜索空间是排列树(图 3-6-2)。图 3-6-2 有 24 个叶片，每个叶片表达一种排列方案。“碰壁回头”意味着无须每条路都走到底，节省搜索的时间，从搜索树中剪去一些无需搜索的边。所以也是一种剪枝的办法。剪去的边越多越节省时间。图 3-6 1 中从(1)到(8)的转移过程可以看作是从搜索树图 3-6 2 的对应分枝作如图所示的剪枝(见图 3 6-3)。虚线表示被剪去的部分， → 表示搜索点的“转移”。或用“向前走碰壁回头”的策略将图 3 6 1 的搜索过程用图 3-6-4 表示， → 表示后退的过程，括弧里的数是搜索的次序。

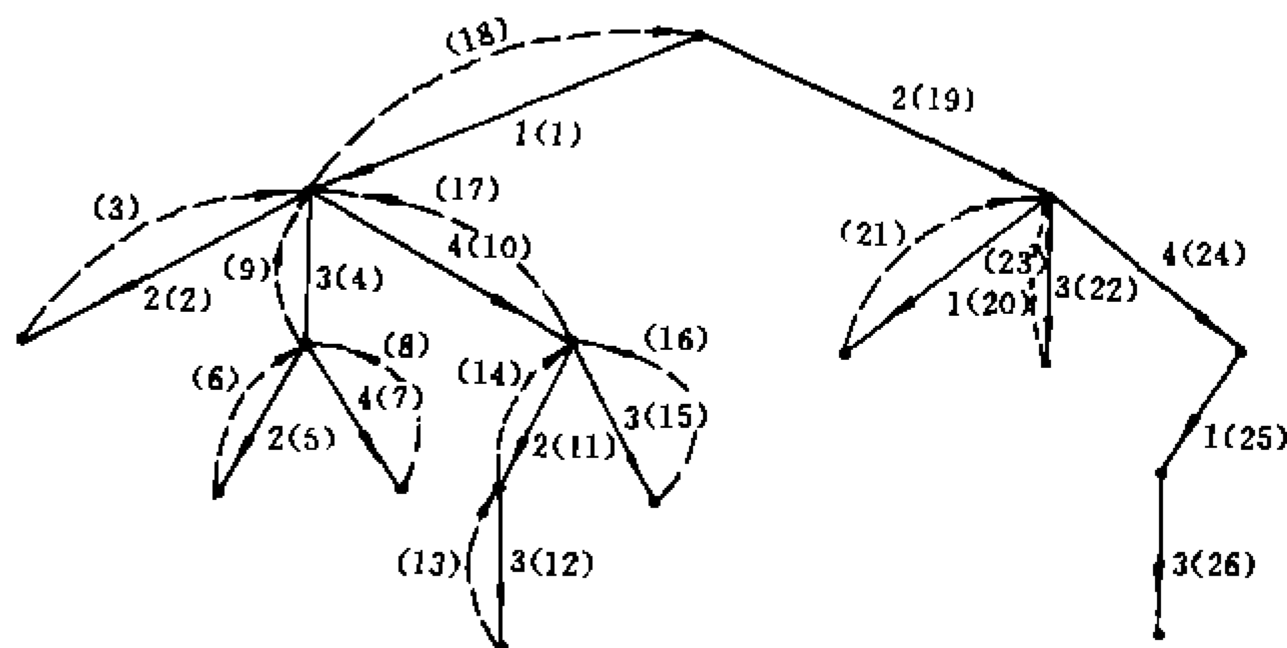


图 3 6 4

现再看一个应用例子：如图 3 6 5 所示，有 12 个域，用 4 种颜色进行着色，要求相邻的域没有相同的颜色，试问有几种着色方案？

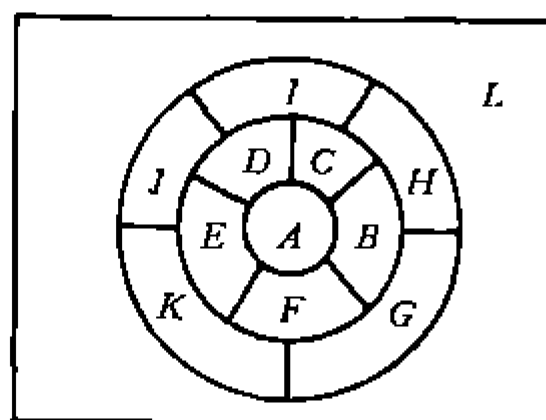


图 3 6-5

设四种颜色分别为 1，2，3，4，而域 A,B,C 先分别着以 1，2，3 三种颜色。对于 A,B,C 以外各域，一一进行着色尝试，从 1 色开始看看是否合适，不合适就改为 2 色，依次类推。对每个域四种颜色都要试过，直到全部试完为止。

这个算法的搜索过程可描述如下：

(1) $c(1)\leftarrow 1$，$c(2)\leftarrow 2$．$c(3)\leftarrow 3$；

i 从 4 到 12 作 $c(i)\leftarrow 1$；　　　　/＊初始化＊/

$i\leftarrow 4$。

(2) $j\leftarrow c(i)$，

若 $j<5$ 则转(3)

否则作【$c(i)\leftarrow 1$，$i\leftarrow i-1$，

若 $i<4$，则转(5)

否则作【$c(i)\leftarrow c(i)+1$，转(2)】】。

(3) k 从 1 到 i　1 作

【若 $c(k) * A(i,k)=j$，

则作【 $c(i) \leftarrow c(i)+1$，转(2)】】。

(4) $i \leftarrow i+1$

若 $i<13$ 则转(2)。

(5) 输出矩阵 C

$c(12) \leftarrow c(12)+1$，

$i \leftarrow 12$，转(2)。

算法中 $c(1:12)$ 矩阵记录颜色；

$$A=(a_{ij})_{2\times12} \text{ 是区域的邻接矩阵；}$$

上面的例子介绍的搜索方法，它的基本思想是一直往前搜索，直到走不通了才后退一步。开始时让 A,B,C 域分别着以颜色 1,2,3。从域 D 开始，先试以颜色 1，看是否发生冲突，若发生冲突则换下一种颜色，直到用完 4 种颜色为止。若颜色用完了，要考虑后退。第二步中当 $j=5$ 时就是这种情形。第二步就是判断当 $C(i)=j$ 时，1 到 $i-1$ 已着色的域与域 i 是否发生冲突，若有冲突，就要更换下一种颜色，否则向前推进。

二、DFS 图的算法

图的 DFS 算法的基本思想是：

已知图 $G=(V,E)$，对图的每一条边进行搜索时：

(1) 当 $E(G)$ 的所有边未经完全搜索时，任取一顶点 $v_i \in V(G)$，给 v_i 以标志且入栈；(以先入后出为原则叫做栈，先入先出者叫队)。

(2) 对与 v_i 点关联的边进行搜索时，若存在另一端点未给标志的边时，则作【把另一端点作为 v_i，给以标志，并且入栈，再转(2)，】，否则转(3)。

(3) 当与 v_i 关联的边全部搜索完毕时(即不存在以 v_i 为端点而未经搜索的边时)，则 v_i 点从栈顶退出，若栈不空则作【即让取走 v_i 后的栈顶元素作为 v_i，转(2)】，否则转(4)。

(4) 若栈已空，但还存在未给标志的顶点时，取其中任一顶点作 v_i，转(2)。若所有顶点都已给标志时，则算法终止。

例 1：图 3-6-6 的邻接矩阵为：

$$A=\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{matrix}\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

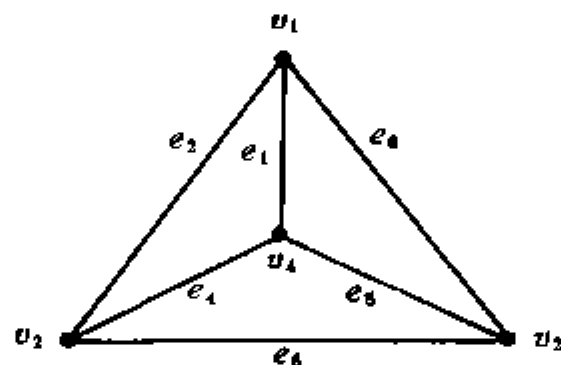


图 3-6-6

设从 v_1 点开始，给 v_1 以标志，与 v_1 相邻的顶点依次为 $\{v_2,v_3,v_4\}$；即

$$Adj(v_1)=\{v_2,v_3,v_4\}。$$

由于第一个邻接点 v_2 未给标志，故 v_2 入栈，且给标志。但 $Adj(v_2)=\{v_1,v_3,v_4\}$，而第一个邻接顶点 v_1 已给标志，故取 (v_2,v_3) 边，给 v_3 以标志且入栈。

又 $Adj(v_3)=\{v_1,v_2,v_4\}$，由于 v_1,v_2 都已给标志，故取 (v_3,v_4) 边，给 v_4 以标志且入栈，但与 v_4 相邻接的顶点全部给了标志，故退栈：此时栈顶为 v_3，但与 v_3 相邻的顶点均已

给标志，故退栈。v_2，v_1 因类似理由依次退栈。此时栈已空。且所有顶点都给以标志，故结束。

上面介绍的是图的 DFS 算法的一般步骤和原则，还是那么一句话“向前走，碰壁回头”。在这过程中留下必要的信息，比如“标志”等，下面利用这原理结合具体问题展开如下：

(a) 无向图的 DFS 算法

在讨论无向图的 DFS 算法前先引进几个必要的概念。DFS 搜索最先开始的点叫做根结点。从 v 点沿 (v,w) 边继续搜索，v 点称为 w 点的“父亲”结点，w 点为 v 点的“儿子”结点，(v,w) 边称之为树枝。当然不言而喻 w 点未给标志，否则不能沿 (v,w) 边向前搜索。若 (v,w) 边的另一端点已给标志，则称 (v,w) 边为后退边。

算法步骤：

(1) TREE←∅，BACK←∅，i←1，

∀ $v \in V$ 作

【father(v)←0，mark(v)←0，】。

(2) 任选一点 r 满足条件 mark(r)＝0，作【v←r，mark(v)←1，num(v)←i，】。

(3) 若所有与 v 点关联的边均已给标志，则转(5)；否则任选一未给标志的边 (v,w) 转(4)。

(4) 给 (v,w) 边以从 v 到 w 的方向，并给以标志 * 以示通过检查。

若 mark(w)＝0 则作

【i←i＋1，num(w)←i，TREE←TREE ∪ {(v,w)}，mark(w)←1，father(w)←v，v←w，转(3)，】。

若 mark(w)＝1 则作

【BACK←BACK ∪ {(v,w)}，转(3)，】。

(5) 若 father(v)≠0 则作

【v←father(v)，转(3)】，

否则作

【若∀ $v \in V$ 恒有 mark(v)＝1 则结束，

否则作

【i←i＋1，转(2)，】。

其中 TREE，BACK 顾名思义分别是树枝边和后退边集合，mark(v)＝0 表示 v 点未经过搜索，mark(v)＝1 则表示 v 已经过搜索。father(v)为 v 点的“父亲”结点。DFS 搜索法搜索的结果产生一棵树或若干棵树。依结点 v 在搜索过程中被访问的先后顺序给以序号 num(v)，即 num(v)表 v 点的 DFS 序号。

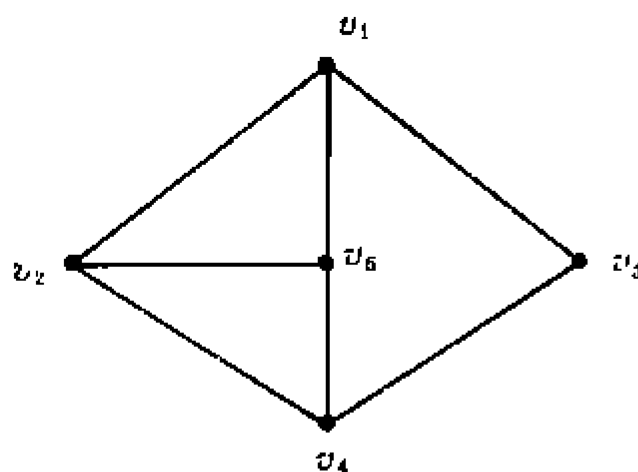


图 3.6.7

例：如图 3.6.7 所示。

$$
A=\begin{bmatrix}
0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1 & 0
\end{bmatrix}
\begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix}
$$
$$\qquad v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5$$

利用上述的 DFS 算法可得如图 3-6-8 所示的有向树(实线)和余树(虚线),箭头指明搜索的过程。

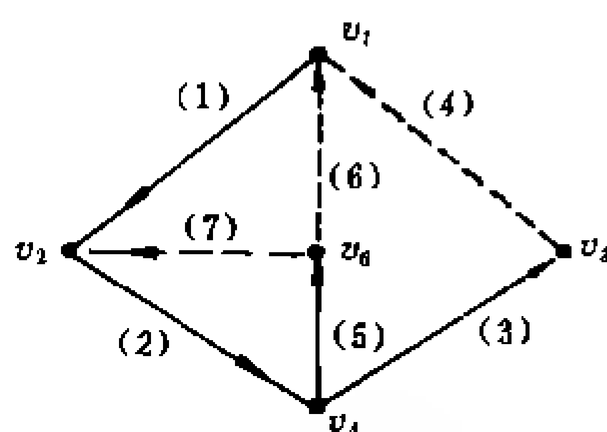


图 3-6-8

图 3-6-8 中各边都有( ),( )里的数表示在搜索的过程中边的次序。搜索过程可形象地表达如图 3-6-9

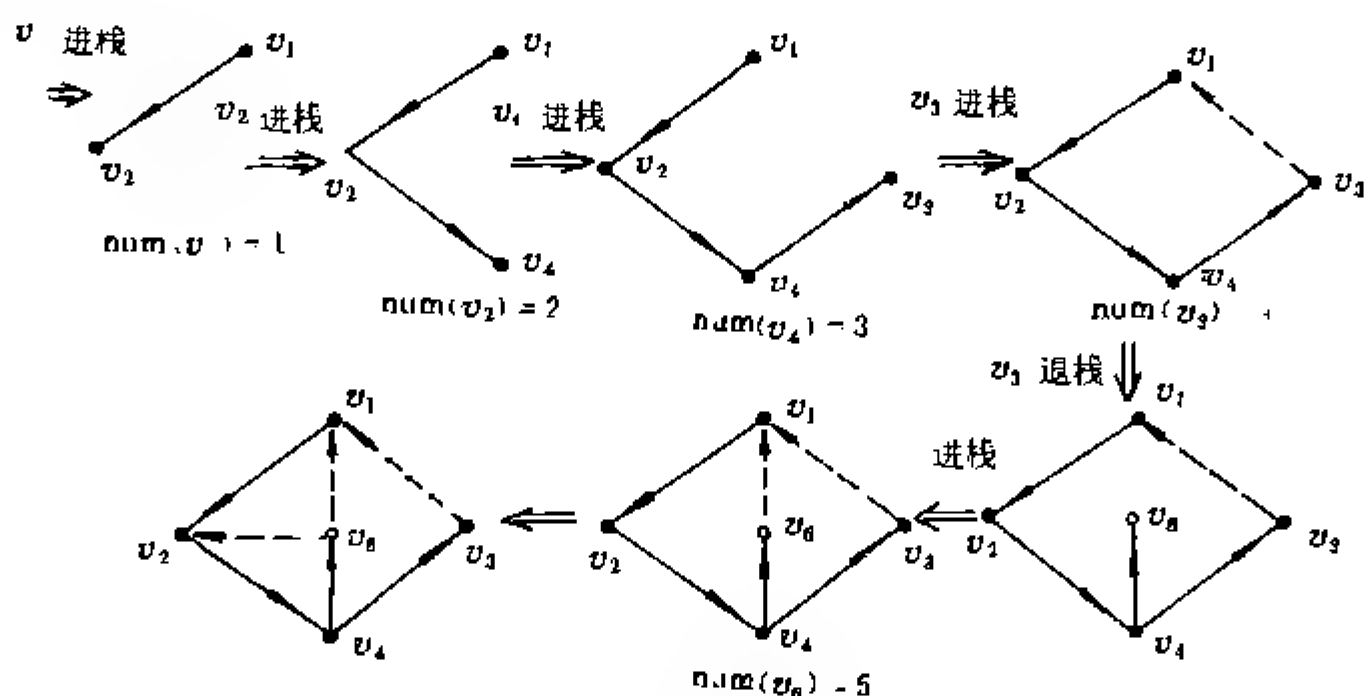


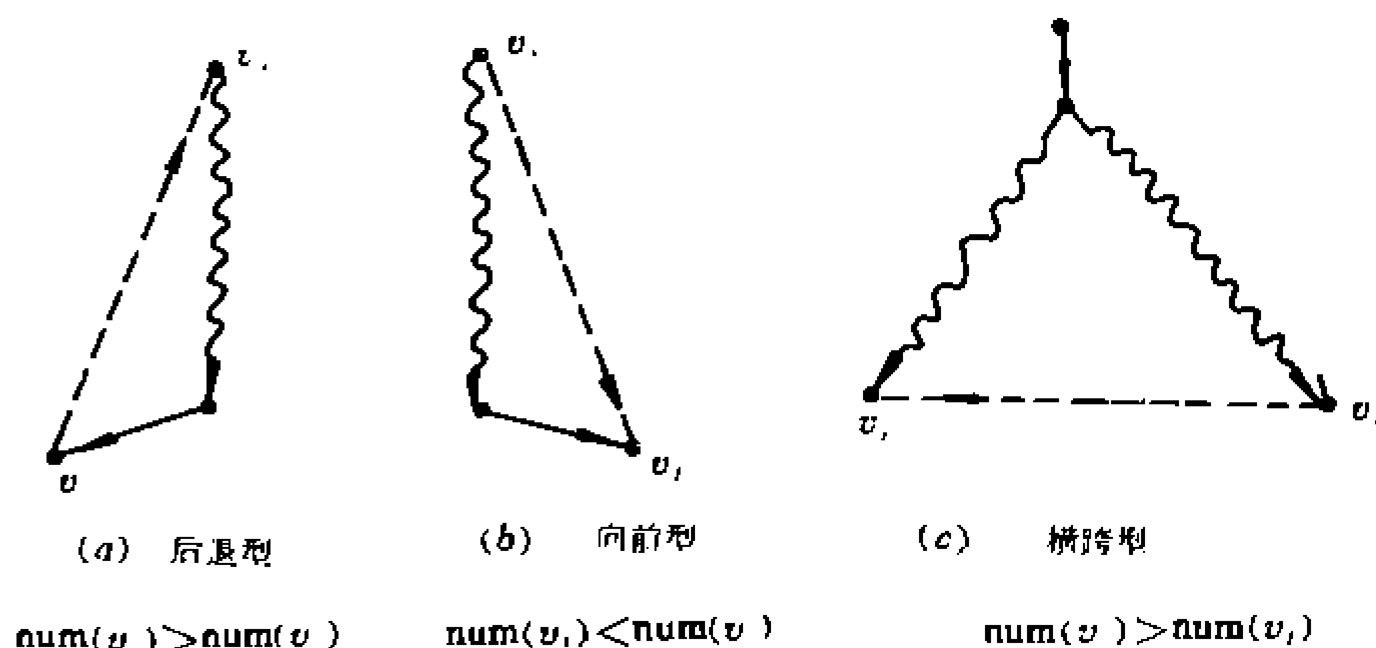
图 3-6-9

利用 DFS 算法于无向图生成的生成树有一特点,它的后退边的方向是从 num 的值高的顶点指向 num 值较小的顶点,即若边(v_i, v_j)是余树边,则 $num(v_i) > num(v_j)$。若从 v_k 点可沿着 DFS 外向树树枝走到 v_t 点时,称 v_k 点是 v_t 点的“祖先”,而 v_t 点是 v_k 点的“子孙”或“后裔”。显然有:$num(v_k) < num(v_t)$。

(*b*) 有向图的 DFS 算法

前面介绍过无向图的 DFS 算法，稍加修改便可用于有向图。

首先要分析清楚有向图不同于无向图之点，无向图 G 通过 DFS 搜索结果，将边分成属于 TREE 和 BACK 两类，属于 TREE 的边 (v,w)，有 $num(v)<num(w)$，而属于 BACK 的边 (v,w) 则 $num(v)>num(w)$。而有向图则情况比较复杂。首先 $num(v)<num(w)$ 的边 (v,w) 不一定都属于 TREE 的，其中还有一种 w 已经搜索过的向前边 (v,w) 的集合，用 FORWARD 来表示它。类似的道理 $num(v)>num(w)$ 的边 (v,w) 也不一定都是一种 BACK，还要区分 v 是否为 w 的"后裔"结点，即从 w 沿 DFS 树的边可到达 v 点的为一类，用 BACK 表示它，还有一种虽然 $num(w)<num(v)$，但 v 不是 w 的"后裔"结点的，即 w 已从栈中撤走，也就是说与 w 点关联的所有的边都已搜索完毕，这样的边用 CROSS 表示称之为横跨边的。横跨边和后退边的区别在于 w 点是和否已经结束搜索，这一点如何刻划？在下面算法中引进了 $scan(v)$ 它们间的区别见图 3-6-10。



(*a*) 后退型 $num(v)>num(v)$　　(*b*) 向前型 $num(v_i)<num(v)$　　(*c*) 横跨型 $num(v)>num(v_i)$

图 3-6-10

还有一点必需注意的，连通的无向图，必然只有一棵 DFS 树。然而连通的有向图可能产生 DFS 森林。DFS 森林中两棵树间只能有属于 CROSS 的边相连，不可能存在属于 FORWARD 的边。这些道理都是显而易见的。

有向图 DFS 算法和无向图十分接近。

(1) TREE←∅，FORWARD←∅，BACK←∅，CROSS←∅，i←1，

$\forall v\in \bar{V}$ 作

【mark(v)←0，father(v)←0，scan(v)←0.】。

(2) 任选 $r\in \bar{V}$，满足条件 mark(r)=0，作

【v←r，mark(v)←1，num(v)←i，】。

(3) 与 v 点关联的所有的边若均已给标志，则作【scan(v)←1，转(5)】，否则任选一未给标志的边 (v,w)，转(4)。

(4) 给 (v,w) 边以标志 *，

若 mark(w)=0 则作

【$i \leftarrow i+1$, num(w)$\leftarrow i$, TREE$\leftarrow$TREE$\cup$ $\{(v,w)\}$, mark(w)$\leftarrow 1$, father(w)$\leftarrow v$, $v \leftarrow w$, 转(3),】, 否则

若 num(v)<num(w)则作

【FORWARD$\leftarrow$FORWARD$\cup\{(v,w)\}$, 转(3),】。

若 num(v)>num(w)且 scan(w)=0 则作

【BACK$\leftarrow$BACK$\cup\{(v,w)\}$, 转(3),】。

若 num(v)>num(w)且 scan(w)=1, 则作

【CROSS$\leftarrow\{(v,w)\}$, 转(3),】。

(5) 若 father(v)$\neq$0, 则作

【$v \leftarrow$ father(v), 转(3),】。否则作

若$\forall u \in \bar{V}$, maR(u)=1, 则结束, 否则作

【$i \leftarrow i+1$, 转(2),】。

例 2:

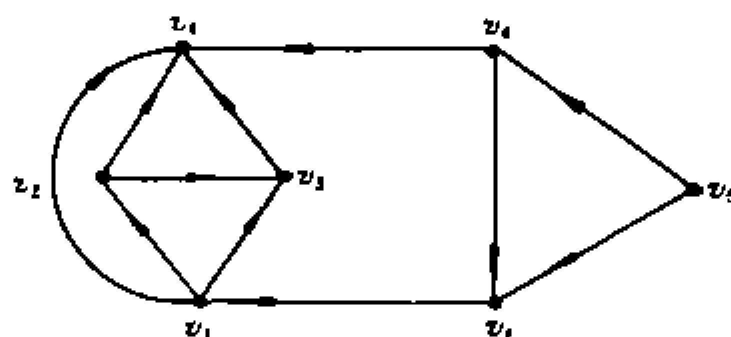


图 3-6-11

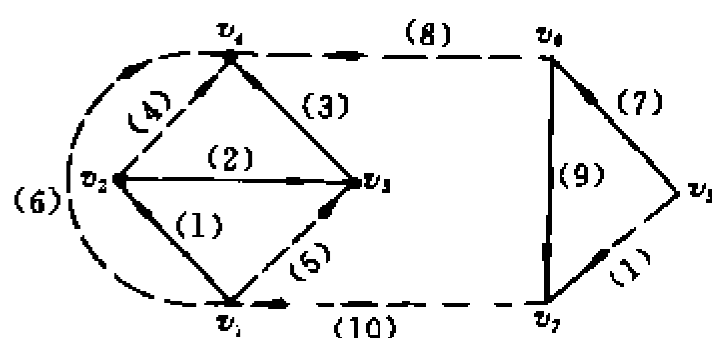


图 3-6-12

用 DFS 法于有向图 G(图 3-6-11), 得一外向森林(图 3-6-12), 各边( )里的数是在搜索过程中该边的先后序号。需要着重指出的有:

(1) 若有存在于不同的两棵树之间的边 $e_k=(v_i,v_j)$, 则 e_k 边的方向是从序号 num 值较大的顶点走向 num 值较小的顶点; 即若(v_i,v_j)是这样的边, 则必然有 num(v_i)>num(v_j), 而且 v_i 点所在的树上的点的序号都大于 v_j 点所在的树上点的序号, 就是前面说的属于 CROSS 的边。虽然 num(v_i)>num(v_j), 它们之间不存"祖先"与"后裔"的关系。

(2) 同一棵树上的两个顶点 v_i、v_j, 必有共同的"祖先", 如果其中一点 v_i 是 v_j 的"祖先", 则它们的一个共同"祖先"便是 v_i 自身。否则如若 v_i 不是 v_j 的祖先, 则 v_i、v_j 的共同"祖先"中必有一个为 v_k 的, 它的 num 值为最小, 即 num(v_k)小于其它共同"祖先"的 num 值。

对 DFS 算法, 我们就简单介绍这些性质, 它有许多用处, 下面所要讲的图的块划分和强连通块的划分就是 DFS 算法在图上的直接应用。进一步的结果可参看相关资料。

§7 图的块划分

先引进一个概念"割切点", 若 $v \in G$, 将 v 点及与 v 点关联的所有的边从 G 中消去, 余下的图的连通块数目增加了, 则称 v 点为割切点。若图 G 是连通图, 从图 G 中消去 v 点及

与 v 点关联的边后余的图为非连通的，则称 v 点是割切点。

当然这里"消去与 v 点关联的边"并不意味着消去另一端点。

无割切点的连通图叫做互连通图，图 G 的最大的互连通图叫做互连通块。

下面假定图 $G=(V,E)$ 是连通的无向图，T 是 DFS 树，r 是它的根节点，如何找出图 G 的割切点呢？首先将割切点的特征找到。

如若根节点 r 是割切点，显然对于 DFS 树 T，当且仅当 r 有多于 1 个"儿子"结点，$v(\neq r)$ 是割切点，当且仅当 v 的任一儿子结点 w 都不存在它的"后裔"（包括 w 在内）结点到 v 的"祖先"结点间的后退边。也就是 w 或 w 的"后裔"结点到 v 的任一"祖先"结点间不存在后退边。

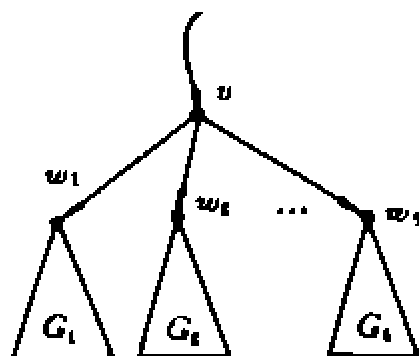


图 3-7-1

这道理是比较明显的。以 $v\neq r$ 为例（见图 3-7-1）。v 有"儿子"节点 $w_1,w_2,\cdots,w_s$。如若 $w_1,w_2,\cdots,w_s$ 都有"后裔"结点到 v 的"祖先"结点间的后退边，则消去 v 点及与 v 点关联的边，不增加连通块的数目。反之，若存在一"儿子"节点，设为 w_i，不存在从 w "后裔"（包含 w 点在内）结点到 v 的"祖先"结点间的边，则消去 (v,w) 边及 v 点，至少子图 G_i 将脱离出来。为此引进

$$\mathrm{low}(v)=\min_{w\in T(v)}\{\mathrm{num}(w)\},$$

其中 $T(v)$ 是由 v 出发沿 DFS 树 T 到达 v 的"后裔"结点 u，最多通过一后退边 (u,w) 到达的点 w 的集合。v 属于 $T(v)$。

low(v)的计算步骤：

1. 第一次访问 v 点时，令 $\mathrm{low}(v)\leftarrow\mathrm{num}(v)$；

2. 遇到后退边 (v,w) 时，令

$$\mathrm{low}(v)\leftarrow\min\{\mathrm{num}(w),\mathrm{low}(v)\}。$$

3. v 的"儿子"节点 w 搜索结束从栈中退出，返回到 v 点时，令

$$\mathrm{low}(v)\leftarrow\min\{\mathrm{low}(v),\mathrm{low}(w)\}。$$

利用 DFS 搜索法对图 $G=(V,E)$ 的全部顶点和边进行检查过程，沿途留下 low(v) 的信息。搜索结束时，$v(\neq r)$ 点是 G 的割切点的充要条件是 v 有"儿子"w 使得

$$\mathrm{low}(w)\geqslant\mathrm{num}(v)。$$

利用 DFS 求割切点的算法：

(1) $\forall\ v\in V$ 作

【 $\mathrm{father}(v)\leftarrow0$, $\mathrm{mark}(v)\leftarrow0$】，

$i\leftarrow1$, STACK$\leftarrow\varnothing$。

(2) 选择 $r\in V$，满足条件 $\mathrm{mark}(r)=0$，

$v\leftarrow r$, $\mathrm{num}(v)\leftarrow(i)$, $\mathrm{low}(v)\leftarrow i$, $\mathrm{mark}(v)\leftarrow1$。

(3) 若与 v 点关联的所有边均已给标志时，转(5)，否则作【任选一未给标志的边(v,w)，给边(v,w)以标志，并将(v,w)边加入到栈 STACK 顶上，转(4)】。

(4) 若 $\mathrm{mark}(w)=0$ 则作

【 $i\leftarrow i+1$, $\mathrm{num}(w)\leftarrow i$, $\mathrm{low}(w)\leftarrow i$, $\mathrm{father}(w)\leftarrow v$, $\mathrm{mark}(w)\leftarrow1$, $v\leftarrow w$, 转(3)】。

若 mark(w)=1 则作

【 low(v)←min{low(v), num(w)},转(3)】。

(5) 若 father(v)≠0 则作

【 若 low(v)≥num(father(v))则作

【 从栈 STACK 中移走边(father(v),v)及其上栈中元素并输出,转(6),】,否则转(6)】,否则结束。

(6) low(father(v))←min{low(v), low(father(v))},

v←father(v),转(3)。

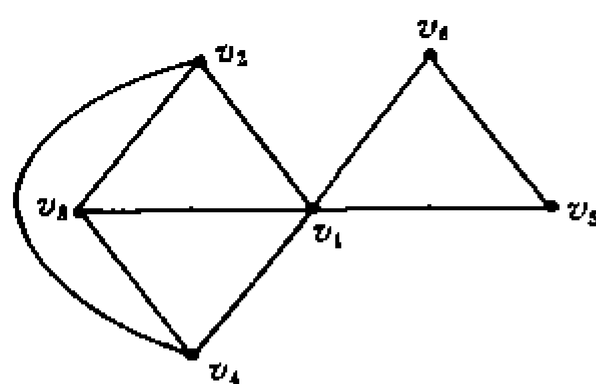


图 3-7-2

例:以图 3-7-2 为例求其割切点,过程见图 3-7-3。

假定从 v_1 出发开始搜索。v_1 点是根结点有两个"儿子"结点,所以是割切点。

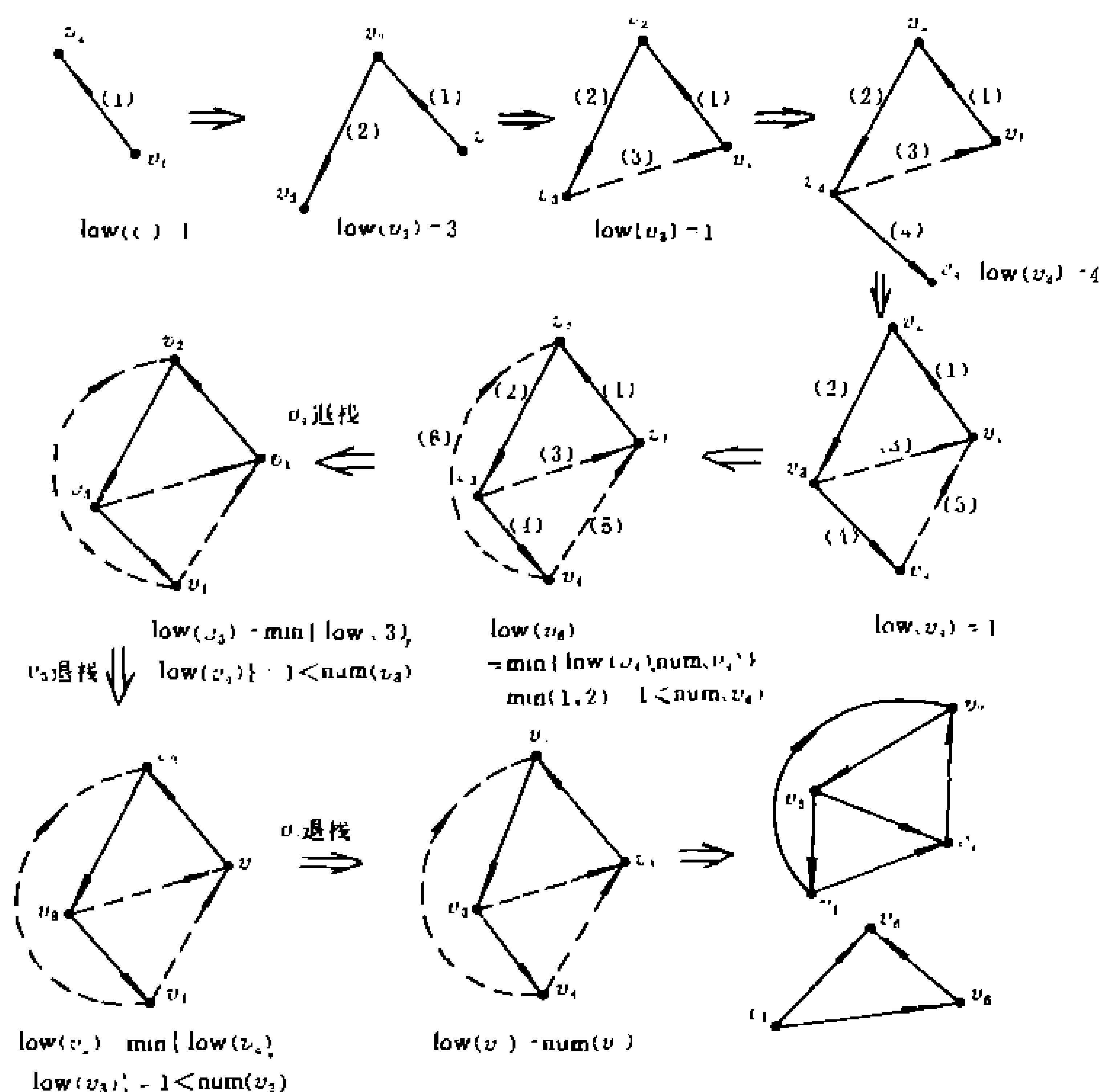


图 3-7-3

§8 强连通块的划分

强连通概念前面已见过，极大的强连通子图叫做强连通块。

利用 DFS 算法在划分强连通块也是非常有效的。已知有向图 $G=(V,E)$，通过 DFS 算法可得一有向森林，这一点已在前面讨论过了。

现代化城市的街道多是单向的，也就是说是有向的单行道。街道和它们的交叉点构成的有向图必然是强连通的。否则 $u \to v$ 存在一条道路，而 v 回到 u 却走不通。

在一计算机系统中，资源用点来表示它。有一程序 p_1 占有资源 s_1，而对 s_2 提出申请。可表以从点 s_1 引向 s_2 的有向边，边 (s_1,s_2) 附以权 p_1。任一瞬间计算机资源的状态图，就是以代表资源的 $s_1,s_2,\cdots,s_m$ 顶点的有向图 G，图 G 的强连通块反映一种死锁现象。最简单的死锁现象比如程序 p_1 占有 s_1 对 s_2 提出申请，而 p_2 占有 s_2 对 s_3 提出要求，而 p_3 占有 s_3 对 s_1 提出要求，结果只能是"你等我，我等你"，互相等待。这就是死锁现象。这是操作系统要避免出现的事件。

首先要对强连通块的构造有一直观的认识。为简单起见假定 G 图有两个强连通子图 G_1 和 G_2，图 G 有一 DFS 森林 T，T 和 G_1，G_2 的交分别是 G_1，G_2 的 DFS 树 T_1 和 T_2，T_1 和 T_2 的根结点分别用 r_1 和 r_2 表之。

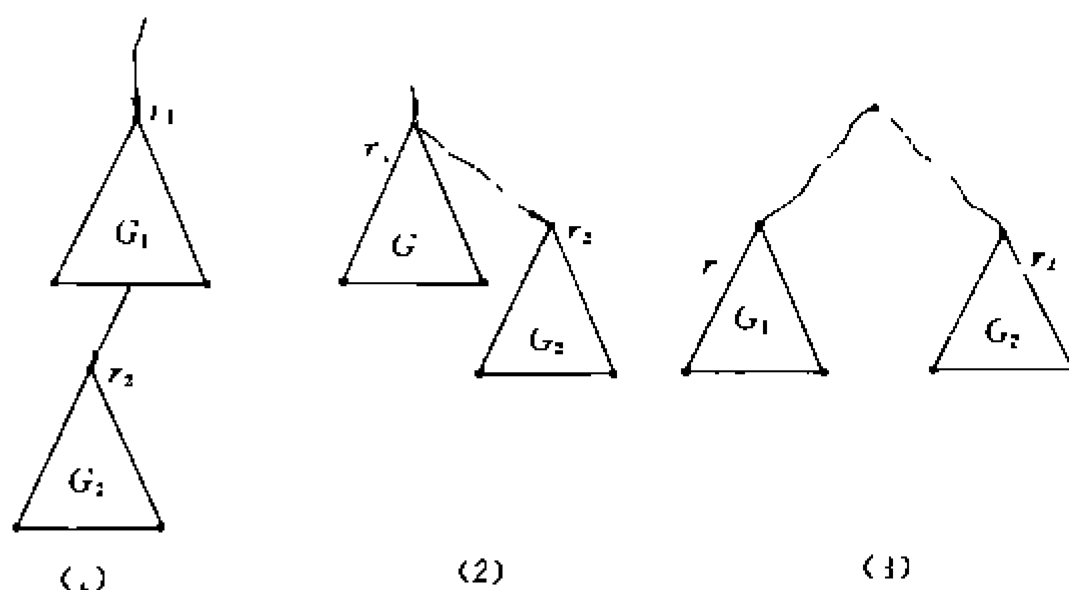


图 3-8-1

从图 3-8-1 中的(1)可知，若 G_2 有一后退边返回到 G_1，则 G_1 和 G_2 将构成一强连块。即以 G_1 和 G_2 为子强连通块的强连通块。同样的情况将发生在 3-8-1(2)中，若 G_2 有一横跨边指向 G_1 时，如图 3-8-2 所示。

但图 3-8-1(3)中，若 G_2 图有一横跨边指向 G_1，却不可能发生上面所说的形成更大的强连通块的情况。

如何判别它们之间的区别呢？为此引进

$$\text{lowlink}(v)=\min_{w \in T_2(v)}\{\text{num}(w)\}。$$

其中 $T_2(v)$ 是从 v 出发沿 DFS 树，最多通过一条后退边或横跨边所能到达的点的集合。v_s

点本身当然也在 $T_2(v)$ 内。

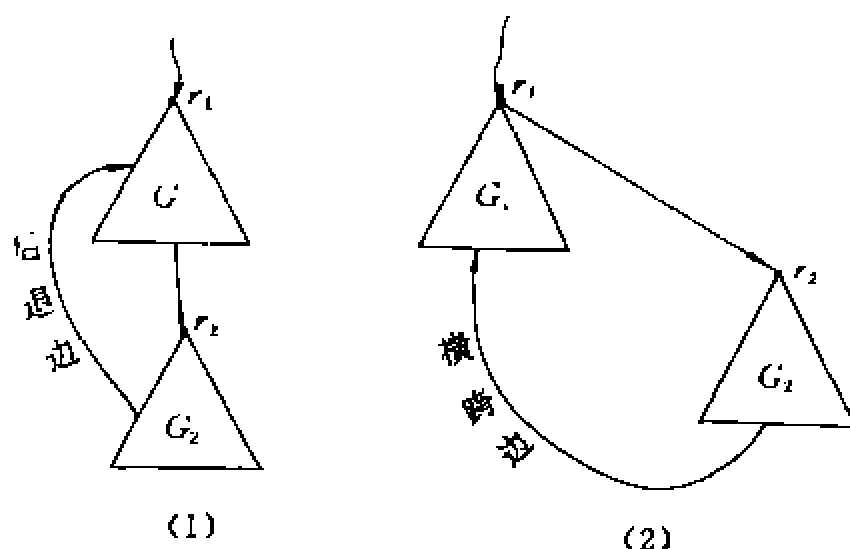


图 3-8 2

lowlink(·)的计算步骤：

(1) 第一次访问 v 点时

$$lowlink(v)\leftarrow num(v);$$

(2) 遇到与 v 点关联的后退边 (v,w) 时

$$lowlink(v)\leftarrow \min\{lowlink(v),num(w)\};$$

(3) 遇到横跨边 (v,w) 时，

$$lowlink(v)\leftarrow \min\{lowlink(v),num(w)\};$$

(4) 若 v 的"儿子" w 搜索完毕，从 w 回到 v 时

$$lowlink(v)\leftarrow \min\{lowlink(v),lowlin(w)\}。$$

v 是强连通块的根结点的充要条件是 $lowlink(v)=num(v)$。

求强连通块的 DFS 算法：

(1) $\forall v\in V$ 作

【 $mark(v)\leftarrow 0$, $father(v)\leftarrow 0$, $star(v)\leftarrow 0$】, $i\leftarrow 1$, $STACK\leftarrow\varnothing$。

(2) 任选一 $mark(r)=0$ 的点 $r\in V$，作

【 $v\leftarrow r$, $num(v)\leftarrow i$, $lowlink(v)\leftarrow i$, v 进栈 STACK, $star(v)\leftarrow 1$】。

(3) 若所有与 v 关联的边均已有标志，则转(5)，否则任选一未标志的边 (v,w)，给以标志 *，转(4)。

(4) 若 $mark(w)=0$ 则作

【 $i\leftarrow i+1$, $num(w)\leftarrow i$, $lowlink(w)\leftarrow i$, $father(w)\leftarrow v$, $mark(w)\leftarrow 1$, w 进栈 STACK, $star(w)\leftarrow 1$, $v\leftarrow w$, 转(3)】，否则作

【 若 $num(w)<num(v)$ 且 $star(w)=1$

则作

【 $lowlink(v)\leftarrow \min\{lowlink(v),num(w)\}$，转(3)】】。

(5) 若 $lowlink(v)=num(v)$ 则作【从 STACK 栈中移出从 v 开始一直到栈顶的元素，并输出，对输出的所有元素 w，作 $star(w)\leftarrow 0$,】,

(6) 若 father(v)=0 则作

【若所有的 $v\in V$, mark(v)=1。则停止，否则作

【lowlink(father(v))←min{lowlink(father(v)), lowlink(v)},

v←father(v), 转(3)】。

star(v)=0(或1)分别标志着 v 点是(否)不在栈 stack 中。

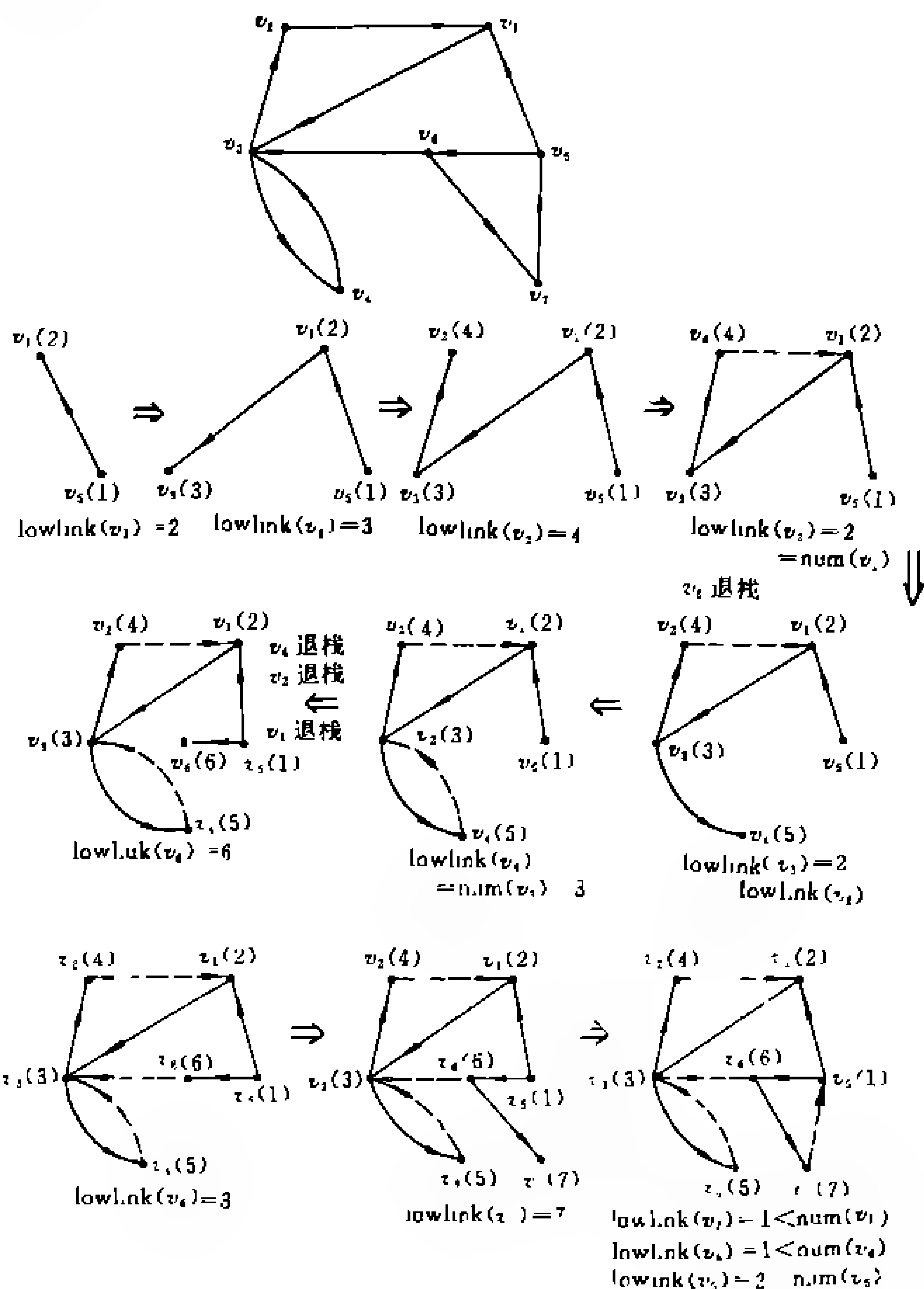


图 3-8-3

现举例说明如下(图3-8-3)。从 v_5 出发利用 DFS 法搜索故存在两个强连通部分：

最后得两个强连通块如图3.8.4所示。

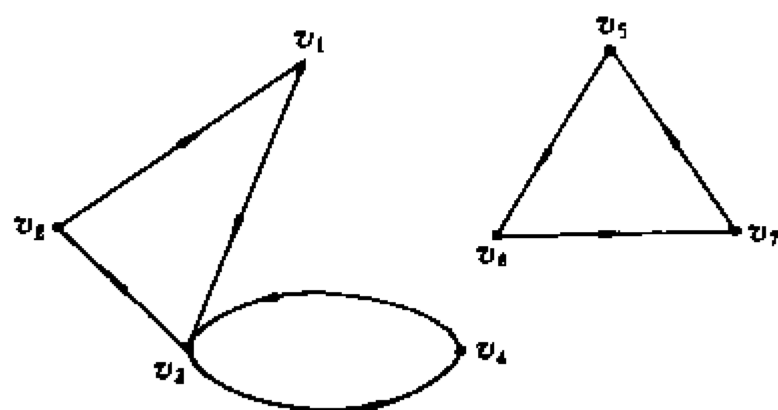


图 3.8.4

习　题

1. 试设计Kruskal算法中判断回路的方法。
2. 试设计一种算法，求图$G=(V,E)$的各顶点到某一顶点的最长距离。
3. 求下图的各顶点到v_1点的最短路径。

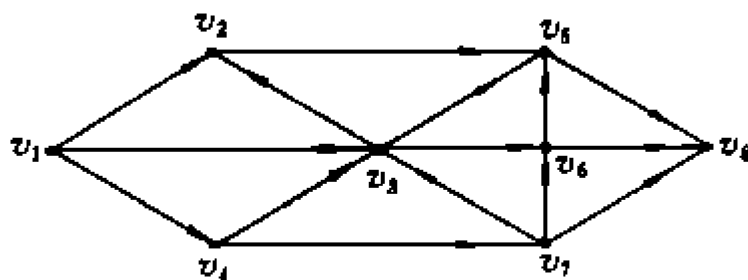


图　习题3

4. 上题的图中去掉各边的方向，求其最短树。
5. 求下图中Z点与$\bar{Z}$点之间的最短路径。

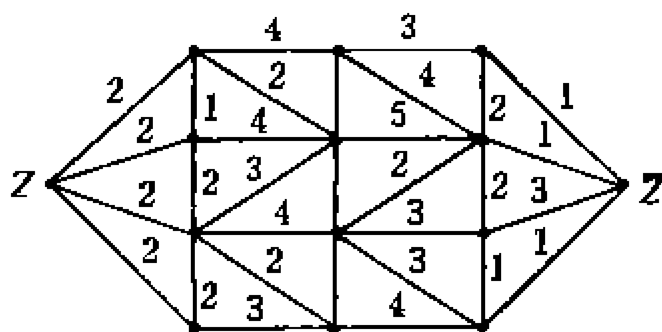


图　习题5

第四章　电路网络问题

图论是六，七十年代发展比较迅速的数学分支，在电路网络的研究方面的应用非常成功。特别是随着科学技术的发展，电路结构越来越复杂，它的设计与计算更多依赖于计算机，图论在其中扮演着重要的角色。它的发展改变了电路网络这门学科的面貌。本章重点讲解图在电路网络中的应用。

§1　克希荷夫定律

假设某一电路网络的图 G 有 n 个顶点 m 条边，

$$\boldsymbol{B} = (b_{ij})_{n\times m},$$

$$\boldsymbol{C} = (c_{ij})_{(m-n+1)\times m}$$

分别是它的关联矩阵和回路矩阵。令

$$\boldsymbol{I} = \begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_m \end{bmatrix}, \qquad \boldsymbol{V} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

分别为这电路的各边电流与各边电压。则我们可得到：

(a) 克希荷夫电流定律

对于每一结点，流入该点电流的代数和为零；即

$$\sum_{k=1}^{m} b_{jk} i_k(t) = 0,\ j = 1,2,\cdots,n。$$

或简单地写成：

$$\boldsymbol{BI} = \boldsymbol{O}。$$

(b) 克希荷夫电压定律

沿着任一回路 $\boldsymbol{C}$，电压降的代数和为零，即

$$\sum_{k=1}^{m} c_{jk} v_k(t) = 0,\ j = 1,2,\cdots,m-n+1。$$

或写成

$$\boldsymbol{CV}_e = \boldsymbol{O}。$$

§2　电路问题

根据克希荷夫定律可知：任何回路中的电压降的总和等于外加电压。设通过线圈的电压降为 u_L，通过电容 C 的电压降为 u_c，通过电阻 R 的电压降为 u_R，分别有：

$$u_L = L\frac{di}{dt},$$

$$u_C = \frac{1}{C}\int_0^t i\,dt.$$

$$u_R = iR,$$

$$\therefore \quad L\frac{di}{dt} + Ri + \frac{1}{C}\int i\,dt = e(t)\text{。}$$

另有电路的初始状态：

$$i\big|_{t=0} = i_0\text{。}$$

若引进 Laplace 变换：

$$L[i(t)] = I(p).$$

则有：

$$L\left[\int_0^t i\,dt\right] = \frac{1}{p}I(p) \quad i.$$

$$L\left[\frac{di}{dt}\right] = pI(p)\text{。}$$

图 4-2-1

令 $L[e(t)] = E(p)$ 则有

$$\left(Lp + R + \frac{1}{Cp}\right)I = E(p) + i$$

令

$$Z_p = Lp + R + \frac{1}{Cp}.$$

称 Z_p 为这电路的运算阻抗。特别当 $i_0 = 0$ 时有：

$$E(p) = Z(p)I(p)$$

即运算电压等于运算阻抗与运算电流的乘积，称之为广义的欧姆定律。

不难证明关于运算阻抗的串并联公式：

(a) 串联公式：如图 4-2-2 所示有：

$$Z(p) = Z_1(p) + Z_2(p).$$

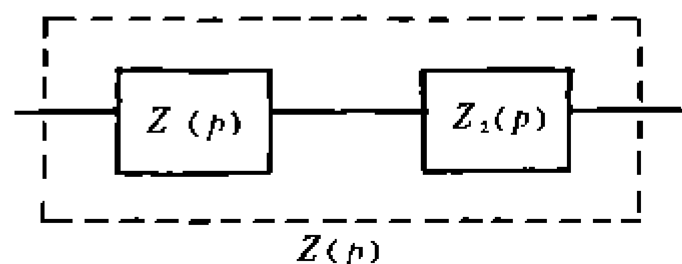


图 4-2-2

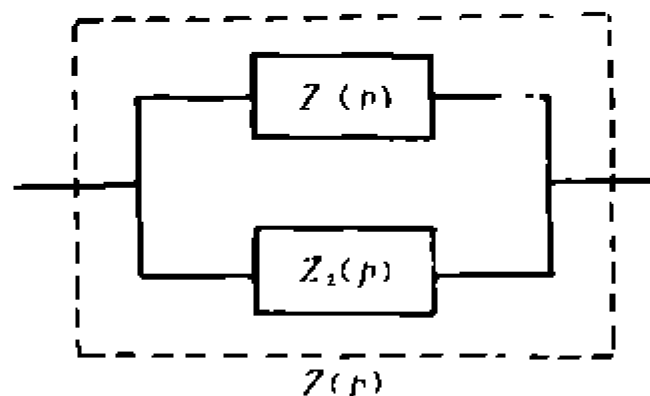


图 4-2-3

(b) 并联公式：如图 4-2-3 所示有：

$$\frac{1}{Z} = \frac{1}{Z_1} + \frac{1}{Z_2}\text{。}$$

证明的方法与串并联公式类似，不再赘述。

综上所述电路问题归纳为：

(1) 克希荷夫电流定律：

$$BI = O_{\circ}$$

(2) 克希荷夫电压定律：

$$CV_e = O_{\circ}$$

(3) 广义的欧姆定律：

$$V_e(p) = Z(p)I(p) + E(p)_{\circ}$$

其中边电流 $i(t)$ 的 Laplace 变换 $I(p)$ 为 m 维列向量，$V_e(p)$ 为边电压的 Laplace 变换，也是 m 维列向量。

$$Z(p) = (z_{ij}(p))_{m\times m}$$

称为阻抗矩阵。当这样的电路网络有 m 条边时，共有 $2m$ 个变量 $I(p)$、$V(p)$。上面 $BI=O$ 有 $n-1$ 个独立方程，$CV_e=O$ 有 $m-n+1$ 个独立方程，再加上广义的欧姆定律的 m 个方程，共有 $2m$ 个方程，联立解出 $I(p)$，$V(p)$，再作 Laplace 反变换给出电路的解。但当 m 充分大时(在现代技术中这是很可能的问题)便十分复杂。下面要引进状态变量法。

§3 状态变量法理论基础

上节已经讨论了具有 m 条边的电路网络，一般有 $2m$ 个变量，即边电流与边电压。但应注意到这样的事实，关于电感 L 所在的边上边电压与边电流之间有关系：

$$u_L = L\frac{di}{dt}_{\circ}$$

而电容 C 所在的边有关系：

$$u_C = \frac{1}{C}\int_{t_0}^{t} i\,dt_{\circ}$$

或

$$i = C\frac{du_C}{dt}_{\circ}$$

电阻 R 所在边的关系是：

$$u_R = iR_{\circ}$$

这些等式表示每一边的电流与电压之间的依赖关系。

另一方面，若对边的次序作适当的安排，使得

$$C_f = (\underbrace{I_{(m-n+1)}}_{m-n+1} \ \vdots\ \underbrace{C_{12}}_{n-1})$$

相应地有

$$B_k = (\underbrace{B_{11}}_{m-n+1}\ \vdots\ \underbrace{B_{12}}_{n-1})_{(n-1)\times m\circ}$$

由 $BI=O$ 得

$$(B_{11} \ \vdots\ B_{12})\begin{pmatrix} I_C \\ I_T \end{pmatrix} = O,$$

其中 I_C 表余树枝上的电流，I_T 表树枝上的电流。

$$\therefore \qquad B_{11}I_C + B_{12}I_T = O,$$

$$I_T = -B_{12}^{-1}B_{11}I_C。 \tag{1}$$

公式(1)告诉我们树枝边上的电流与余树枝边的电流的关系。只要知道余树枝边的电流，则所有的电流都可以计算而得。

另一方面，类似地从 $CV=O$ 得

$$(I_{(m-n+1)} \ \vdots\ C_{12})V = O。$$

即

$$(I_{(m-n+1)} \ \vdots\ C_{12})\begin{pmatrix} V_C \\ V_T \end{pmatrix} = O。$$

其中 V_C、V_T 分别表示余树枝边的电压与树枝边的电压。

所以
$$V_C = -C_{12}V_T,$$

但根据第二章讨论的结果可知：

$$C_{12} = -B_{11}^T(B_{12}^{-1})^T,$$

$$\therefore \qquad C_{12}^T = -B_{12}^{-1}B_{11}。$$

令
$$K = -C_{12} = B_{11}^T(B_{12}^{-1})^T.$$

可得

$$V_C = KV_T,$$

$$I_T = -K^TI_C$$

即

$$\begin{pmatrix} V_C \\ I_T \end{pmatrix} = \begin{pmatrix} O & K \\ -K^T & O \end{pmatrix}\begin{pmatrix} I_C \\ V_T \end{pmatrix},$$

其中
$$K = (k_{ij})_{(m-n+1)\times(n-1)}。$$

这个公式说明了 m 条边的电流和电压，只要知道其中的余树枝的电流和树枝的电压便可求出其它。这个公式在状态变量法中扮演着重要的角色，它可以起减少独立变量的作用。

§4 状态变量法

上节得出一个重要公式

$$\begin{pmatrix} V_C \\ I_T \end{pmatrix} = \begin{pmatrix} O & K \\ -K^T & O \end{pmatrix}\begin{pmatrix} I_C \\ V_T \end{pmatrix}。$$

下面先通过一个例子，说明如何利用上面的式子把电路问题化为一常微分方程组，然后把它抽象成一般的方法。

例 1：

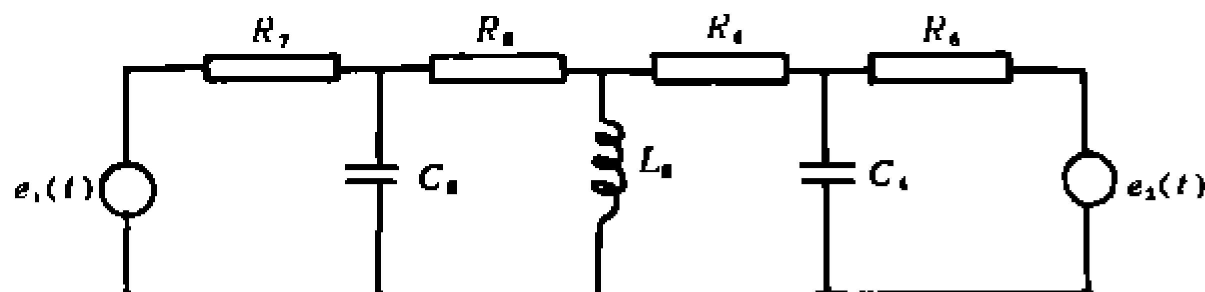


图 4-4-1

图 4-4-1 的电路网络对应的图是图 4-4-2。

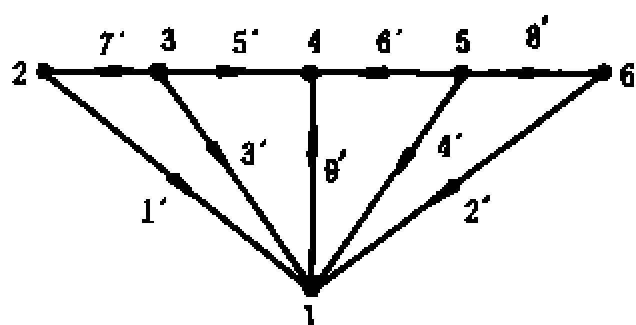


图 4-4-2

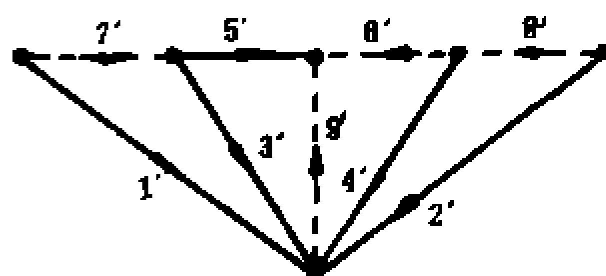


图 4-4-3

图论中常用的符号 v 和 e 在电路中有其它的意义，为了避免发生混乱，故图 4-4-2 中用 1，2，…，表示顶点，而用 1′，2′，…，表示边。这一章后面均如此，作为一种约定。

图 4-4-2 中有一棵树 T，它的树枝包含了所有的电容、电压源和部分电阻。如图 4-4-3 中实线所示（图中虚线为对应的余树枝边）。

其中

$$
\boldsymbol{V}_C=\begin{pmatrix} v_6 \\ v_7 \\ v_8 \\ v_9 \end{pmatrix}=\begin{pmatrix} R_6 i_6 \\ R_7 i_7 \\ R_8 i_8 \\ L_9 \dfrac{di_9}{dt} \end{pmatrix}, \qquad \boldsymbol{I}_C=\begin{pmatrix} i_6 \\ i_7 \\ i_8 \\ i_9 \end{pmatrix},
$$

$$
\boldsymbol{I}_T=\begin{pmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{pmatrix}=\begin{pmatrix} i_1 \\ i_2 \\ C_3\dfrac{dv_3}{dt} \\ C_4\dfrac{dv_4}{dt} \\ G_5 v_5 \end{pmatrix}, \qquad \boldsymbol{V}_T=\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix}=\begin{pmatrix} e_1(t) \\ e_2(t) \\ v_3 \\ v_4 \\ v_5 \end{pmatrix}.
$$

$$
\boldsymbol{B}=\begin{matrix}1\\2\\3\\4\\5\\6\\ \end{matrix}\begin{bmatrix}1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1\\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}_{\begin{matrix}1' & 2' & 3' & 4' & 5' & 6' & 7' & 8' & 9'\end{matrix}},
$$

$$
\boldsymbol{B}=\left[\begin{array}{ccccc:cccc}1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1\\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0\end{array}\right]_{\begin{matrix}1' & 2' & 3' & 4' & 5' & 6' & 7' & 8' & 9'\end{matrix}}
$$

$$
=\begin{bmatrix}0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1\\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1\\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}_{\begin{matrix}6' & 7' & 8' & 9' & 1' & 2' & 3' & 4' & 5'\end{matrix}},
$$

$$
\boldsymbol{B}_{12}=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 1\\ 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0\end{bmatrix},
$$

$$
\boldsymbol{B}_{2}'=\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1\\ 0 & 1 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & -1 & 0 & 0\end{bmatrix},
$$

$$
\boldsymbol{K}=\boldsymbol{B}^{T}_{1}(\boldsymbol{B}_{2}')^{T}=\begin{bmatrix}0 & 0 & 1 & 1 & 0\\ 1 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 1\\ 0 & 0 & -1 & 0 & 0\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0 & 1\\ 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0\end{bmatrix}
$$

$$
=\begin{bmatrix}0 & 0 & -1 & 1 & 1\\ 1 & 0 & -1 & 0 & 0\\ 0 & 1 & 0 & -1 & 0\\ 0 & 0 & 1 & 0 & 1\end{bmatrix}。
$$

以之代入公式：

$$\begin{pmatrix} V_C \\ I_T \end{pmatrix} = \begin{pmatrix} O & K \\ -K^T & O \end{pmatrix} \begin{pmatrix} I_C \\ V_T \end{pmatrix}.$$

则

$$\begin{pmatrix} R_6 i_6 \\ R_7 i_7 \\ R_8 i_8 \\ L_9 \dfrac{di_9}{dt} \\ i_1 \\ i_2 \\ C_3 \dfrac{dv_3}{dt} \\ C_4 \dfrac{dv_4}{dt} \\ G_5 v_5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_6 \\ i_7 \\ i_8 \\ i_9 \\ e_1(t) \\ e_2(t) \\ v_3 \\ v_4 \\ v_5 \end{pmatrix}.$$

展开得：

$$L_9 \frac{di_9}{dt} = v_5 - v_3, \tag{1}$$

$$C_3 \frac{dv_3}{dt} = i_6 + i_7 + i_9, \tag{2}$$

$$C_4 \frac{dv_4}{dt} = i_8 - i_6, \tag{3}$$

$$v_5 = -R_5(i_6 + i_9), \tag{4}$$

$$i_6 = (v_4 - v_3 + v_5)/R_6, \tag{5}$$

$$i_7 = (-v_3 + e_1(t))/R_7, \tag{6}$$

$$i_8 = (e_2(t) - v_4)/R_8, \tag{7}$$

$$i_1 = -i_7, \tag{8}$$

$$i_2 = -i_8. \tag{9}$$

从(4)、(5)得关于变量 v_5、i_6 的方程组：

$$\begin{cases} v_5 + R_5 i_6 = -R_5 i_9, \\ \dfrac{v_5}{R_6} + i_6 = (v_4 - v_3)/R_6. \end{cases}$$

∴

$$v_5 = \begin{vmatrix} -R_5 i_9 & R_5 \\ (v_4 - v_3)/R_6 & 1 \end{vmatrix} \Bigg/ \begin{vmatrix} 1 & R_5 \\ -1/R_6 & 1 \end{vmatrix}$$

$$= \left[-R_5 i_9 + R_5(v_3 - v_4)/R_6\right] \Big/ \left(1 + \frac{R_5}{R_6}\right)$$

$$= \frac{R_5 v_3 - R_5 R_6 i_9 - R_5 v_4}{R_5 + R_6}.$$

$$i_6 = (v_4 - v_3)/R_6 + \frac{R_5 v_3 - R_5 R_6 i_9 - R_5 v_4}{R_6(R_5 + R_6)} = \frac{R_6(v_4 - v_3) - R_5 R_6 i_9}{R_6(R_5 + R_6)}。$$

又 代入(1)得

$$\begin{aligned}
L_9 \frac{di_9}{dt} &= \frac{R_5 v_3 - R_5 R_6 i_9 - R_5 v_4}{R_5 + R_6} - v_3 \\
&= \frac{-R_6 v_3 - R_5 R_6 i_9 - R_5 v_4}{R_5 + R_6} \\
&= (-R_5 R_6 i_9 - R_6 v_3 - R_5 v_4)/(R_5 + R_6) \\
&= \frac{-R_5 R_6}{R_5 + R_6} i_9 - \frac{R_6}{R_5 + R_6} v_3 - \frac{R_5}{R_5 + R_6} v_4。
\end{aligned}$$

类似地从(2)、(3)可得到

$$C_3 \frac{dv_3}{dt} = \frac{R_6}{R_5 + R_6} i_9 - \left(\frac{1}{R_7} + \frac{1}{R_5 + R_6}\right) v_3 + \frac{1}{R_5 + R_6} v_4 + \frac{e_1(t)}{R_7},$$

$$C_4 \frac{dv_4}{dt} = \frac{R_5}{R_5 + R_6} i_9 + \frac{1}{R_5 + R_6} v_3 - \left(\frac{1}{R_8} + \frac{1}{R_5 + R_6}\right) v_4 + \frac{e_2(t)}{R_8}。$$

或写成矩阵形式：

$$\frac{d}{dt}\begin{bmatrix} i_9 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} -\frac{R_5 R_6}{L_9(R_5 + R_6)} & -\frac{R_6}{L_9(R_5 + R_6)} & -\frac{R_5}{L_9(R_5 + R_6)} \\ \frac{R_6}{C_3(R_5 + R_6)} & \frac{-(R_5 + R_6 + R_7)}{C_3(R_5 + R_6)R_7} & \frac{1}{C_3(R_5 + R_6)} \\ \frac{R_5}{C_4(R_5 + R_6)} & \frac{1}{C_4(R_5 + R_6)} & \frac{-(R_5 + R_6 + R_8)}{R_8 C_4(R_5 + R_6)} \end{bmatrix} \begin{bmatrix} i_9 \\ v_3 \\ v_4 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 0 \\ \frac{1}{C_3 R_7} & 0 \\ 0 & \frac{1}{C_4 R_8} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \end{bmatrix}。$$

解这微分方程组，从而得其它边的电流、电压，称 i_9、v_3、v_4 为状态变量。

现在将状态变量法的一般步骤叙述如下：先从网络图中找出一棵树，它的树枝包含了所有的电容、电压源和部分电阻，这棵树称之为正常树。它的余树的边包含了所有的电感、电流源和部分电阻。

令 $\boldsymbol{V}_L$，$\boldsymbol{I}_L$ 分别表示电感 L 所在的边的电压与电流；

$\boldsymbol{V}_{CS}$，$\boldsymbol{I}_{CS}$ 分别表示有电流源的边的电压和电流；

$\boldsymbol{V}_R$，$\boldsymbol{I}_R$ 分别表示有电阻 R 的边的电压和电流；

$\boldsymbol{V}_C$，$\boldsymbol{I}_C$ 分别表示有电容 C 的边的电压和电流；

V_{VS}，I_{VS} 分别表示有电压源边的电压和电流；

V_G，I_G 分别表示有导纳 G 边的电压和电流。

$$
\begin{pmatrix} V_L \\ V_{CS} \\ V_R \\ I_C \\ I_{VS} \\ I_G \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & K_{11} & K_{12} & K_{13} \\ 0 & 0 & 0 & K_{21} & K_{22} & K_{23} \\ 0 & 0 & 0 & K_{31} & K_{32} & K_{33} \\ -K_{11}^T & -K_{21}^T & -K_{31}^T & 0 & 0 & 0 \\ -K_{12}^T & -K_{22}^T & -K_{32}^T & 0 & 0 & 0 \\ -K_{13}^T & K_{23}^T & -K_{33}^T & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} I_L \\ I_{CS} \\ I_R \\ V_C \\ V_{VS} \\ V_G \end{pmatrix}。 \tag{1'}
$$

但从(1')可得：

$$
\begin{cases} V_L = L\dfrac{dI_L}{dt} = K_{11}V_C + K_{12}V_{VS} + K_{13}V_G, \\ I_C = C\dfrac{dV_C}{dt} = -K_{11}^T I_L - K_{21}^T I_{CS} - K_{31}^T I_R。 \end{cases} \tag{2'}
$$

$$
\begin{cases} V_R = RI_R = K_{31}V_C + K_{32}V_{VS} + K_{33}V_G, \\ I_G = GV_G = -K_{13}^T I_L - K_{23}^T I_{CS} - K_{33}^T I_R。 \end{cases} \tag{3'}
$$

$$
\begin{cases} V_{CS} = K_{21}V_L + K_{22}V_{CS} + K_{23}V_G, \\ I_{VS} = -K_{12}^T I_L - K_{22}^T I_{CS} - K_{32}^T I_R。 \end{cases} \tag{4'}
$$

从(2')可知

$$
\frac{dI_L}{dt} = L^{-1}K_{11}V_C + L^{-1}K_{12}V_{VS} + L^{-1}K_{13}V_G,
$$

$$
\frac{dV_C}{dt} = -C^{-1}K_{11}^T I_L - C^{-1}K_{21}^T I_{CS} - C^{-1}K_{31}^T I_R。
$$

令

$$
X = \begin{pmatrix} I_L \\ V_C \end{pmatrix}, \qquad U = \begin{pmatrix} I_{CS} \\ V_{VS} \end{pmatrix},
$$

则有：

$$
\frac{dX}{dt} = \begin{pmatrix} O & L^{-1}K_{11} \\ -C^{-1}K_{11}^T & O \end{pmatrix} X + \begin{pmatrix} O & L^{-1}K_{12} \\ -C^{-1}K_{21}^T & O \end{pmatrix} U + \begin{pmatrix} O & L^{-1}K_{13} \\ -C^{-1}K_{31}^T & O \end{pmatrix} \begin{pmatrix} I_R \\ V_G \end{pmatrix}。
$$

类似办法从(3')可得：

$$
\begin{pmatrix} I_R \\ V_G \end{pmatrix} = \begin{pmatrix} O & R^{-1}K_{31} \\ G^{-1}K_{13}^T & O \end{pmatrix} X + \begin{pmatrix} O & R^{-1}K_{32} \\ G^{-1}K_{23}^T & O \end{pmatrix} U + \begin{pmatrix} O & R^{-1}K_{33} \\ -G^{-1}K_{33}^T & O \end{pmatrix} \begin{pmatrix} I_R \\ V_G \end{pmatrix}。
$$

整理得：

$$
\begin{pmatrix} I_{11} & R^{-1}K_{33} \\ G^{-1}K_{33}^T & I_{22} \end{pmatrix} \begin{pmatrix} I_R \\ V_G \end{pmatrix} = \begin{pmatrix} O & R^{-1}K_{31} \\ -G^{-1}K_{13}^T & O \end{pmatrix} X + \begin{pmatrix} O & R^{-1}K_{32} \\ -G^{-1}K_{23}^T & O \end{pmatrix} U。
$$

$$\therefore \qquad \begin{pmatrix} I_R \\ V_G \end{pmatrix} = DX + EU。$$

其中
$$D = \begin{pmatrix} I_1 & -R^{-1}K_{33} \\ G^{-1}K_{33}^T & I_{22} \end{pmatrix}^{-1} \begin{pmatrix} O & R^{-1}K_{31} \\ -G^{-1}K_{13}^T & O \end{pmatrix},$$

$$E = \begin{pmatrix} I_{11} & R^{-1}K_{33} \\ G^{-1}K_{33}^T & I_{22} \end{pmatrix}^{-1} \begin{pmatrix} O & R^{-1}K_{32} \\ G^{-1}K_{23}^T & O \end{pmatrix}。$$

$$\therefore \qquad \frac{dX}{dt} = MX + NU。$$

其中

$$M = \begin{pmatrix} O & L^{-1}K_1 \\ C^{-1}K_{11}^T & O \end{pmatrix} + \begin{pmatrix} O & L^{-1}K_{13} \\ -C^{-1}K_{31}^T & O \end{pmatrix} D,$$

$$N = \begin{pmatrix} O & L^{-1}K_{12} \\ -C^{-1}K_{21}^T & O \end{pmatrix} + \begin{pmatrix} O & L^{-1}K_{13} \\ -C^{-1}K_{3}^T & O \end{pmatrix} E。$$

另一方面：

$$\begin{pmatrix} V_{CS} \\ I_{VS} \end{pmatrix} = \begin{pmatrix} O & K_{21} \\ -K_{12}^T & O \end{pmatrix} X + \begin{pmatrix} O & K_{22} \\ -K_{22}^T & O \end{pmatrix} U + \begin{pmatrix} O & K_{23} \\ -K_{32}^T & O \end{pmatrix} \begin{pmatrix} I_R \\ V_G \end{pmatrix}$$

$$= \begin{pmatrix} O & K_{21} \\ K_{12}^T & O \end{pmatrix} X + \begin{pmatrix} O & K_{22} \\ K_{22}^T & O \end{pmatrix} U + \begin{pmatrix} O & K_{23} \\ K_{32}^T & O \end{pmatrix} (DX + EU)$$

$$= FX + GU。$$

其中：

$$F = \begin{pmatrix} O & K_{21} \\ -K_{12}^T & O \end{pmatrix} + \begin{pmatrix} O & K_{23} \\ -K_{32}^T & O \end{pmatrix} D,$$

$$G = \begin{pmatrix} O & K_{22} \\ -K_{22}^T & O \end{pmatrix} + \begin{pmatrix} O & K_{23} \\ -K_{32}^T & O \end{pmatrix} E。$$

综上所述，只要解出变量 X，则电路网络中其它变量都可通过它而获得。我们便称变量 X 为状态变量，而求 X 的问题导致解一线性微分方程组。不仅如此，网络系统稳定性判断也容易解决。状态变量法的优点是突出的，它对于利用计算机来设计电路网络起重大的作用。计算机可以根据电路网络的拓扑结构及参数，自动列出方程并给出解答。这个成就是计算机辅助设计的典范。把科技工作者从列方程、编程序的繁琐而缓慢的工作中解放出来，加快了工作速度，提高了效率。特别是促进了利用计算机的模拟或计算机实验取代耗资而又低效的实验室实验。

§5 状态变量法举例

下面举几个状态变量法的例子：

例 1：

图 4-5-1 的电路对应的图如图 4-5-2 所示，其中实线部分为正常树，虚线部分为其对

应余树。

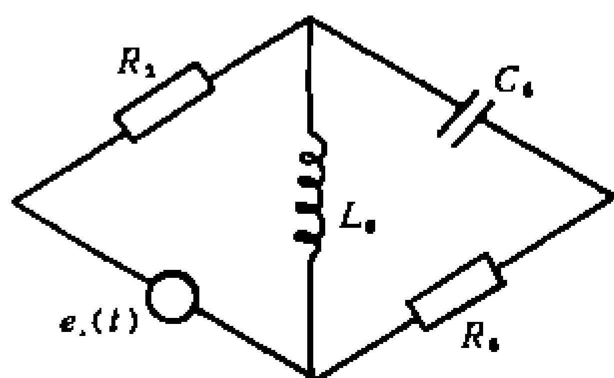


图 4-5-1

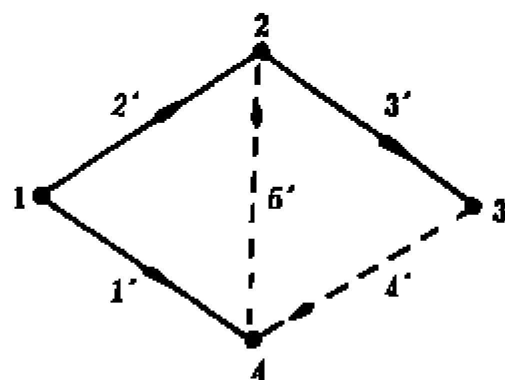


图 4-5-2

$$
\boldsymbol{B}=\begin{array}{cc}\begin{matrix}1\\2\\3\\4\end{matrix}&\begin{pmatrix}0&0&1&1&0\\0&-1&0&-1&1\\1&0&0&0&1\\-1&1&-1&0&0\end{pmatrix}\\ &\begin{matrix}4'&5'&1'&2'&3'\end{matrix}\end{array},
$$

$$
\boldsymbol{B}_f=\begin{pmatrix}0&0&1&1&0\\0&-1&0&1&1\\1&0&0&0&-1\end{pmatrix},
$$

$$
\boldsymbol{B}_{f1}=\begin{pmatrix}0&0\\0&-1\\1&0\end{pmatrix},\qquad \boldsymbol{B}_{f2}=\begin{pmatrix}1&1&0\\0&1&1\\0&0&-1\end{pmatrix}。
$$

$$
\boldsymbol{B}_{f2}^{-1}=\begin{pmatrix}1&1&1\\0&1&-1\\0&0&1\end{pmatrix},\qquad \boldsymbol{K}^T=\boldsymbol{B}_{f2}^{-1}\boldsymbol{B}_{f1}=\begin{pmatrix}1&-1\\-1&1\\1&0\end{pmatrix}。
$$

∵ $\begin{pmatrix}\boldsymbol{V}_l\\\boldsymbol{I}_T\end{pmatrix}=\begin{pmatrix}\boldsymbol{O}&\boldsymbol{K}\\\boldsymbol{K}^T&\boldsymbol{O}\end{pmatrix}\begin{pmatrix}\boldsymbol{I}_l\\\boldsymbol{V}_T\end{pmatrix}$

$$
\boldsymbol{V}_l=\begin{pmatrix}v_4\\v_5\end{pmatrix},\ \boldsymbol{I}_T=\begin{pmatrix}i_1\\i_2\\i_3\end{pmatrix},\ \boldsymbol{I}_l=\begin{pmatrix}i_4\\i_5\end{pmatrix},\ \boldsymbol{V}_T=\begin{pmatrix}v_1\\v_2\\v_3\end{pmatrix}。
$$

∴ $$
\begin{pmatrix}v_4\\v_5\\i_1\\i_2\\i_3\end{pmatrix}=\begin{pmatrix}0&0&1&-1&-1\\0&0&-1&1&0\\-1&1&0&0&0\\1&1&0&0&0\\1&0&0&0&0\end{pmatrix}\begin{pmatrix}i_4\\i_5\\v\\v_2\\v_3\end{pmatrix}。
$$

∵ $v_4=R_4i_4$，　$v_5=L_5\dfrac{di_5}{dt}$，

$i_2=v_2/R_2$。　$i_3=C_3\dfrac{dv_3}{dt}$。

$$\therefore \quad \begin{pmatrix} R_4 i_4 \\ L_5 \dfrac{di_5}{dt} \\ i \\ v_2/R_2 \\ C_3 \dfrac{dv_3}{dt} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i \\ i_5 \\ e_1(t) \\ v_2 \\ v_3 \end{pmatrix}。$$

展开得

$$L_5 \frac{di_5}{dt} = v_2 - e_1(t), \tag{1}$$

$$C_3 \frac{dv_3}{dt} = i_4, \tag{2}$$

$$R_4 i_4 = - v_2 - v_3 + e_1(t), \tag{3}$$

$$\frac{v_2}{R_2} = i_4 - i_5。 \tag{4}$$

(3)、(4)看作是变量 v_2、i_4 的方程组，解得：

$$i_4 = \begin{vmatrix} -v_3 + e(t) & 1 \\ i_5 & -1/R_2 \end{vmatrix} \Bigg/ \begin{vmatrix} R_4 & 1 \\ 1 & 1/R_2 \end{vmatrix}$$

$$= \frac{i_5 + v_3/R_2 - e(t)/R_2}{1 + R_4/R_2} = \frac{R_2 i_5 - v_3 + e_1(t)}{R_2 + R_4}。$$

$$\therefore \quad \frac{dv_3}{dt} = -\frac{1}{(R_2+R_4)C_3} v_3 + \frac{R_2}{(R_2+R_4)C_3} i_5 + \frac{1}{(R_2+R_4)C_3} e(t)。$$

类似可得

$$\frac{di_5}{dt} = \frac{R_2}{L_5(R_2+R_4)} v_3 - \frac{R_2 R_4}{L_5(R_2+R_4)} i_5 + \frac{R_4 e(t)}{L_5(R_2+R_4)}。$$

上面可写成

$$\frac{d}{dt}\begin{pmatrix} i_5 \\ v_3 \end{pmatrix} = \begin{pmatrix} \dfrac{-R_2 R_4}{L_5(R_2+R_4)} & \dfrac{R_2}{L_5(R_2+R_4)} \\ \dfrac{R_2}{C_3(R_2+R_4)} & \dfrac{-1}{C_3(R_2+R_4)} \end{pmatrix} \begin{pmatrix} i_5 \\ v_3 \end{pmatrix} + \begin{pmatrix} \dfrac{R_4}{L_5(R_2+R_4)} \\ \dfrac{1}{C_3(R_2+R_4)} \end{pmatrix} e_1(t),$$

i_5、v_3 是这电路网络的状态变量。

例 2：

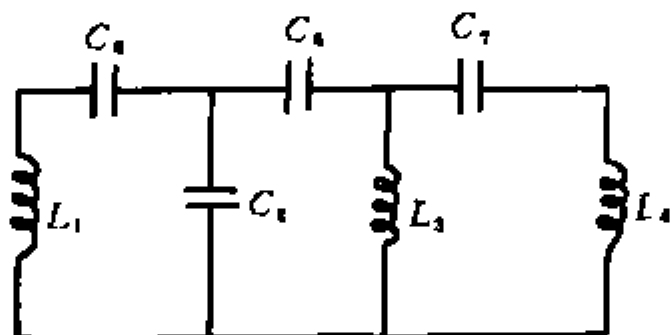


图 4-5-3

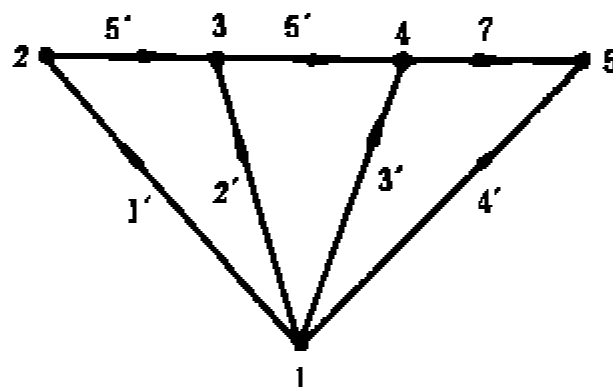


图 4-5-4

图 4 5 3 的电路的图如图 4 5-4 所示。

由含电容的边 2′、5′、6′、7′ 组成正常树如图 4-5 5 所示。

$$
\boldsymbol{B}=\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \ \end{matrix}\begin{matrix}\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 \end{bmatrix} \\ \begin{matrix} 1' & 2' & 3' & 4' & 5' & 6' & 7' \end{matrix}\end{matrix},
$$

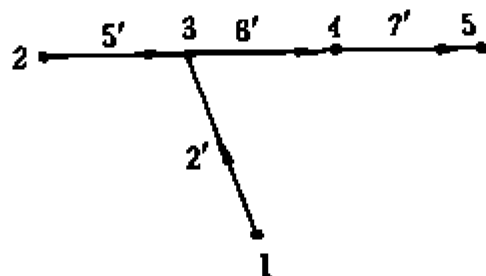


图 4-5-5

$$
\boldsymbol{B}_t=\begin{matrix} 2 \\ 3 \\ 4 \\ 5 \\ \ \end{matrix}\begin{matrix}\left[\begin{array}{ccc:cccc} -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{array}\right] \\ \begin{matrix} 1' & 3' & 4' & 2' & 5' & 6' & 7' \end{matrix}\end{matrix},
$$

$$
\boldsymbol{B}_{11}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix},\qquad \boldsymbol{B}_{12}=\begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix},
$$

$$
\boldsymbol{B}_{12}^{-1}=\begin{bmatrix} -1 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}。
$$

$$
\therefore\quad \boldsymbol{K}^T=-\boldsymbol{B}_{12}^{-1}\boldsymbol{B}_{11}=-\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}=-\begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},
$$

$$
\boldsymbol{V}_L=\begin{bmatrix} v_1 \\ v_3 \\ v_4 \end{bmatrix}=\begin{bmatrix} L_1\dfrac{di_1}{dt} \\ L_3\dfrac{di_3}{dt} \\ L_4\dfrac{di_4}{dt} \end{bmatrix},\qquad \boldsymbol{I}_T=\begin{bmatrix} i_2 \\ i_5 \\ i_6 \\ i_7 \end{bmatrix}=\begin{bmatrix} C_2\dfrac{dv_2}{dt} \\ C_5\dfrac{dv_5}{dt} \\ C_6\dfrac{dv_6}{dt} \\ C_7\dfrac{dv_7}{dt} \end{bmatrix}
$$

$$
\boldsymbol{I}_L=\begin{bmatrix} i_1 \\ i_3 \\ i_4 \end{bmatrix},\qquad \boldsymbol{V}_T=\begin{bmatrix} v_2 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix}。
$$

以之代入公式：

$$
\begin{bmatrix} \boldsymbol{V}_L \\ \boldsymbol{I}_T \end{bmatrix}=\begin{bmatrix} \boldsymbol{O} & \boldsymbol{K} \\ -\boldsymbol{K}^T & \boldsymbol{O} \end{bmatrix}\begin{bmatrix} \boldsymbol{I}_L \\ \boldsymbol{V}_T \end{bmatrix},
$$

得

$$\begin{pmatrix} L_1\dfrac{di_1}{dt} \\ L_3\dfrac{di_3}{dt} \\ L_4\dfrac{di_4}{dt} \\ C_2\dfrac{dv_2}{dt} \\ C_5\dfrac{dv_5}{dt} \\ C_6\dfrac{dv_6}{dt} \\ C_7\dfrac{dv_7}{dt} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_1 \\ i_3 \\ i_4 \\ v_2 \\ v_5 \\ v_6 \\ v_7 \end{pmatrix},$$

$$\therefore \quad \begin{pmatrix} L_1 & 0 & 0 \\ 0 & L_3 & 0 \\ 0 & 0 & L_4 \end{pmatrix} \begin{pmatrix} \dfrac{di_1}{dt} \\ \dfrac{di_3}{dt} \\ \dfrac{di_4}{dt} \end{pmatrix} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} v_2 \\ v_5 \\ v_6 \\ v_7 \end{pmatrix}。$$

$$\therefore \quad \frac{d}{dt}\begin{pmatrix} i_1 \\ i_3 \\ i_4 \end{pmatrix} = \begin{pmatrix} 1/L_1 & 0 & 0 \\ 0 & 1/L_3 & 0 \\ 0 & 0 & 1/L_4 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} v_2 \\ v_5 \\ v_6 \\ v_7 \end{pmatrix}$$

$$= \begin{pmatrix} 1/L_1 & -1/L_1 & 0 & 0 \\ 1/L_3 & 0 & 1/L_3 & 0 \\ 1/L_4 & 0 & 1/L_4 & 1/L_4 \end{pmatrix} \begin{pmatrix} v_2 \\ v_5 \\ v_6 \\ v_7 \end{pmatrix}。$$

类似的办法可得：

$$\frac{d}{dt}\begin{pmatrix} v_2 \\ v_5 \\ v_6 \\ v_7 \end{pmatrix} = \begin{pmatrix} -1/C_2 & -1/C_2 & -1/C_2 \\ 1/C_5 & 0 & 0 \\ 0 & -1/C_6 & -1/C_6 \\ 0 & 0 & 1/C_7 \end{pmatrix} \begin{pmatrix} i_1 \\ i_3 \\ i_4 \end{pmatrix}。$$

合并得方程组：

$$\frac{d\boldsymbol{X}}{dt} = \boldsymbol{MX}。$$

其中

$$
X=\begin{pmatrix} i_1\\ i_3\\ i_4\\ v_2\\ v_5\\ v_6\\ v_7 \end{pmatrix},\quad M=\begin{pmatrix}
0 & 0 & 0 & 1/L_1 & -1/L_1 & 0 & 0\\
0 & 0 & 0 & 1/L_3 & 0 & 1/L_3 & 0\\
0 & 0 & 0 & 1/L_4 & 0 & 1/L_4 & 1/L_4\\
-1/C_2 & -1/C_2 & -1/C_2 & 0 & 0 & 0 & 0\\
-1/C_5 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -1/C_6 & -1/C_6 & 0 & 0 & 0 & 0\\
0 & 0 & -1/C_7 & 0 & 0 & 0 & 0
\end{pmatrix}。
$$

例 3：这电路网络对应的图和它的正常树分别为图 4-5-7 和图 4-5-8。

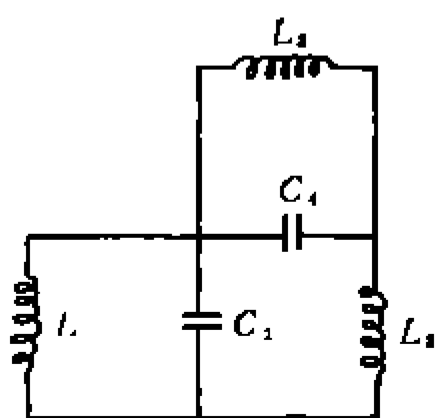


图 4-5-6

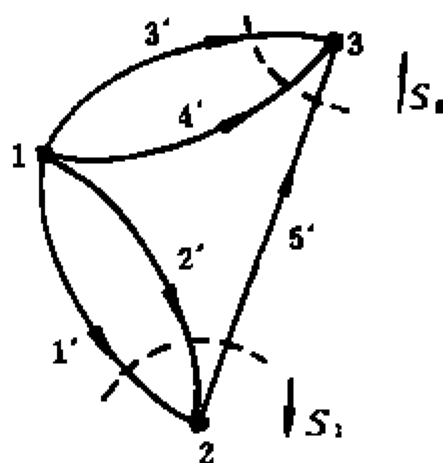


图 4-5-7

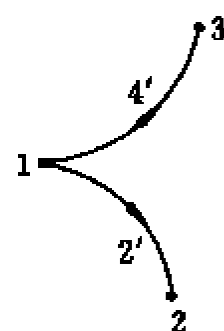


图 4-5-8

$$
B_f=\begin{matrix}1\\2\end{matrix}\begin{pmatrix} 1 & 1 & 0 & \vdots & 1 & 1\\ -1 & 0 & 1 & \vdots & 1 & 0 \end{pmatrix}。
$$
$$
\quad\ 1'\ \ 3'\ \ 5'\ \quad 2'\ \ 4'
$$

$$
B_{11}=\begin{pmatrix} 1 & 1 & 0\\ -1 & 0 & 1 \end{pmatrix},\qquad B_{12}=\begin{pmatrix} 1 & 1\\ -1 & 0 \end{pmatrix}。
$$

$$
\therefore\quad B_{12}^{-1}=\begin{pmatrix} 0 & -1\\ 1 & 1 \end{pmatrix}。
$$

$$
K=B_{11}^T(B_{12}^{-1})^T=\begin{pmatrix} 1 & 1\\ 1 & 0\\ 0 & 1 \end{pmatrix}\begin{pmatrix} 0 & -1\\ -1 & 1 \end{pmatrix}=\begin{pmatrix} 1 & 0\\ 0 & 1\\ -1 & 1 \end{pmatrix}。
$$

另一方面：

$$
S_f=\begin{pmatrix} 1 & 0 & -1 & \vdots & 1 & 0\\ 0 & 1 & 1 & 0 & & 1 \end{pmatrix}\begin{matrix}s_1\\ s_2\end{matrix}
$$
$$
1'\ \ 3'\ \ \ 5'\ \ 2'\ \ 4'
$$

而且从 $K=S_{11}^T$ 可同样给出

$$
K=\begin{pmatrix} 1 & 0\\ 0 & 1\\ -1 & 1 \end{pmatrix}。
$$

将 K 代入

$$\begin{pmatrix} \boldsymbol{V}_C \\ \boldsymbol{I}_T \end{pmatrix} = \begin{pmatrix} \boldsymbol{O} & \boldsymbol{K} \\ -\boldsymbol{K}^T & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \boldsymbol{I}_C \\ \boldsymbol{V}_T \end{pmatrix},$$

其中

$$\begin{pmatrix} \boldsymbol{I}_C \\ \boldsymbol{V}_T \end{pmatrix} = \begin{bmatrix} i_1 \\ i_3 \\ i_5 \\ v_2 \\ v_4 \end{bmatrix}, \qquad \begin{pmatrix} \boldsymbol{V}_C \\ \boldsymbol{I}_T \end{pmatrix} = \begin{bmatrix} v_1 \\ v_3 \\ v_5 \\ i_2 \\ i_4 \end{bmatrix}。$$

但

$$\begin{bmatrix} v_1 \\ v_3 \\ v_5 \end{bmatrix} = \begin{bmatrix} L_1 \dfrac{di_1}{dt} \\ L_3 \dfrac{di_3}{dt} \\ L_5 \dfrac{di_5}{dt} \end{bmatrix} = \begin{bmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ 0 & 0 & L_3 \end{bmatrix} \begin{bmatrix} \dfrac{di_1}{dt} \\ \dfrac{di_3}{dt} \\ \dfrac{di_5}{dt} \end{bmatrix},$$

$$\boldsymbol{I}_T = \begin{bmatrix} i_2 \\ i_4 \end{bmatrix} = \begin{bmatrix} C_2 \dfrac{dv_2}{dt} \\ C_4 \dfrac{dv_4}{dt} \end{bmatrix} = \begin{bmatrix} C_2 & 0 \\ 0 & C_4 \end{bmatrix} \begin{bmatrix} \dfrac{dv_2}{dt} \\ \dfrac{dv_4}{dt} \end{bmatrix},$$

则

$$\begin{pmatrix} \boldsymbol{V}_C \\ \boldsymbol{I}_T \end{pmatrix} = \begin{bmatrix} L_1 & 0 & 0 & 0 & 0 \\ 0 & L_3 & 0 & 0 & 0 \\ 0 & 0 & L_5 & 0 & 0 \\ 0 & 0 & 0 & C_2 & 0 \\ 0 & 0 & 0 & 0 & C_4 \end{bmatrix} \begin{bmatrix} \dfrac{di_1}{dt} \\ \dfrac{di_3}{dt} \\ \dfrac{di_5}{dt} \\ \dfrac{dv_2}{dt} \\ \dfrac{dv_4}{dt} \end{bmatrix}。$$

$$\therefore \quad \frac{d}{dt} \begin{bmatrix} i_1 \\ i_3 \\ i_5 \\ i_2 \\ i_4 \end{bmatrix} = \left[\begin{array}{ccc|cc} L_1 & 0 & 0 & 0 & 0 \\ 0 & L_3 & 0 & 0 & 0 \\ 0 & 0 & L_5 & 0 & 0 \\ \hline 0 & 0 & 0 & C_2 & 0 \\ 0 & 0 & 0 & 0 & C_4 \end{array}\right]^{-1} \left[\begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \\ \hline -1 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \end{array}\right] \begin{bmatrix} i_1 \\ i_3 \\ i_5 \\ v_2 \\ v_4 \end{bmatrix}$$

$$
-\begin{pmatrix}1/L_1 & 0 & 0 & 0 & 0\\ 0 & 1/L_3 & 0 & 0 & 0\\ 0 & 0 & 1/L_5 & 0 & 0\\ 0 & 0 & 0 & 1/C_2 & 0\\ 0 & 0 & 0 & 0 & 1/C_4\end{pmatrix}\begin{pmatrix}0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 1 & 1\\ -1 & 0 & 1 & 0 & 0\\ 0 & 1 & 1 & 0 & 0\end{pmatrix}\begin{pmatrix}i_1\\ i_3\\ i_5\\ v_2\\ v_4\end{pmatrix}
$$

$$
-\begin{pmatrix}0 & 0 & 0 & 1/L_1 & 0\\ 0 & 0 & 0 & 0 & 1/L_3\\ 0 & 0 & 0 & -1/L_5 & 1/L_5\\ -1/C_2 & 0 & 1/C_2 & 0 & 0\\ 0 & /C_4 & -1/C_4 & 0 & 0\end{pmatrix}\begin{pmatrix}i\\ i_3\\ i_5\\ v_2\\ v_4\end{pmatrix}。
$$

例 4：如图 4 5 9(a)所示的电路的对应图为 4-5-9(b)，其正常树为图 4-5-9(c)。

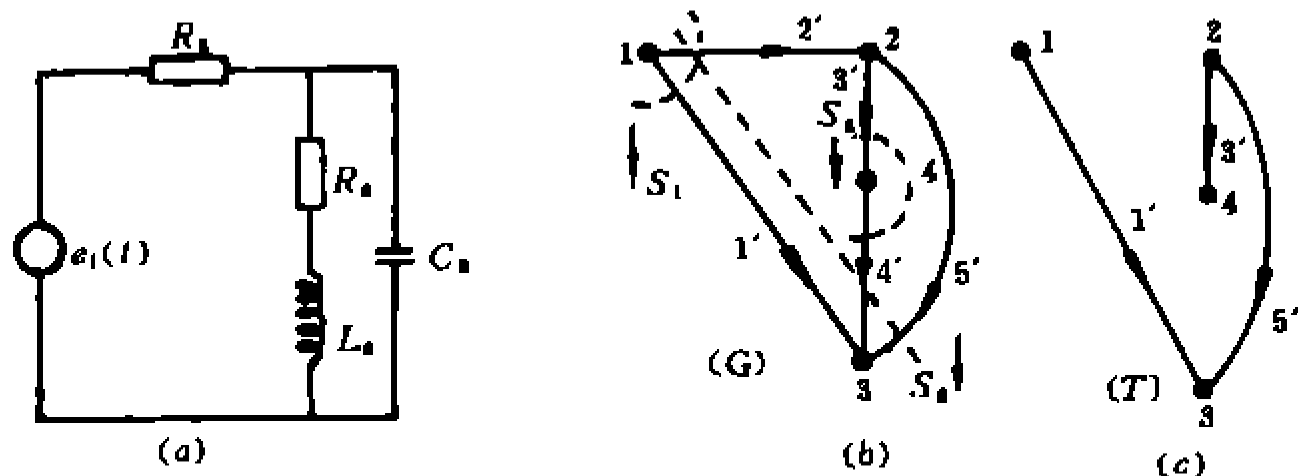


图 4 5 9

$$
B_t\quad\begin{matrix}1\\2\\3\end{matrix}\begin{pmatrix}1 & 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 1 & 1\\ 0 & 1 & 1 & 0 & 1\end{pmatrix},\qquad \begin{matrix}2' & 4' & 1' & 3' & 5'\end{matrix}
$$

$$
B_{11}=\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 1\\ -1 & 0 & 1\end{pmatrix},\qquad B_{12}'=\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 1\\ 1 & 0 & 1\end{pmatrix},
$$

$$
K=B^{1}(B_{12})^T=\begin{pmatrix}1 & -1 & 0\\ 0 & 0 & -1\end{pmatrix}\begin{pmatrix}1 & 1 & 1\\ 0 & 1 & 0\\ 0 & 1 & 1\end{pmatrix}=\begin{pmatrix}1 & 0 & -1\\ 0 & 1 & 1\end{pmatrix}。
$$

另一方面从基本割集矩阵：

$$
S_f=\begin{matrix}S_1\\S_2\\S_3\end{matrix}\left(\begin{array}{cc:ccc}1 & 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 1 & 0\\ 1 & 1 & 0 & 0 & 1\end{array}\right),\qquad \begin{matrix}2' & 4' & 1' & 3' & 5'\end{matrix}
$$

同样可得：

$$K = S_{11}^{T} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & -1 & 1 \end{pmatrix}。$$

由于

$$\begin{pmatrix} V_C \\ I_T \end{pmatrix} = \begin{pmatrix} O & K \\ -K^T & O \end{pmatrix} \begin{pmatrix} I_C \\ V_T \end{pmatrix},$$

$$\begin{bmatrix} v_2 \\ v_4 \\ i_1 \\ i_3 \\ i_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_2 \\ i_4 \\ v_1 \\ v_3 \\ v_5 \end{bmatrix}。$$

但

$$v_4 = L_4 \frac{di_4}{dt} - v_5 - v_3,$$

$$i_5 = C_5 \frac{dv_5}{dt} - i_2 \quad i_4。$$

$$\therefore \quad \frac{d}{dt}\begin{bmatrix} i_4 \\ v_5 \end{bmatrix} - \begin{bmatrix} 0 & \frac{1}{L_4} \\ \frac{-1}{C_5} & 0 \end{bmatrix} \begin{bmatrix} i_4 \\ v_5 \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{L_4} \\ \frac{1}{C_5} & 0 \end{bmatrix} \begin{bmatrix} i_2 \\ v_3 \end{bmatrix}。$$

但

$$v_2 = R_2 i_2 - v_1 \quad v_5,$$

$$i_3 - G_3 v_3 - i_4。$$

$$\therefore \quad \begin{bmatrix} i_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 & \frac{-1}{R_2} \\ \frac{1}{G_3} & 0 \end{bmatrix} \begin{bmatrix} i_4 \\ v_5 \end{bmatrix} + \begin{bmatrix} \frac{1}{R_2} \\ 0 \end{bmatrix} e_1(t)。$$

若令

$$X = \begin{pmatrix} i_4 \\ v_5 \end{pmatrix},$$

则得：

$$\frac{dX}{dt} - \begin{bmatrix} 0 & \frac{1}{L_4} \\ \frac{-1}{C_5} & 0 \end{bmatrix} X + \begin{bmatrix} 0 & \frac{-1}{L_4} \\ \frac{1}{C_5} & 0 \end{bmatrix} \left\{ \begin{bmatrix} 0 & \frac{-1}{R_2} \\ \frac{1}{G_3} & 0 \end{bmatrix} X + \begin{bmatrix} \frac{1}{R_2} \\ 0 \end{bmatrix} e_1(t) \right\},$$

整理得：

$$\frac{dX}{dt} = MX + NU。$$

其中

$$M = \begin{pmatrix} \dfrac{-1}{G_3L_4} & \dfrac{1}{L_4} \\ \dfrac{-1}{C_5} & \dfrac{-1}{R_2C_5} \end{pmatrix}, \quad N = \begin{pmatrix} 0 \\ \dfrac{1}{R_2C_5} \end{pmatrix}, \quad U = e(t)。$$

例 5：

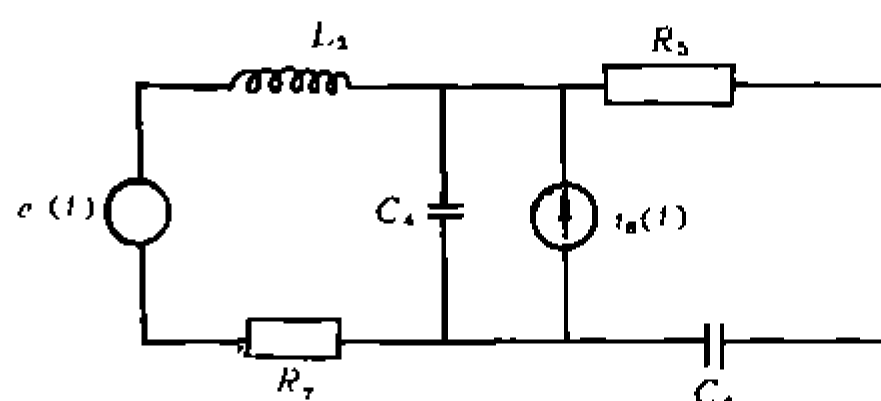


图 4-5-10

上面图 4-5-10 的电路网络的图如下图 4 5 11。

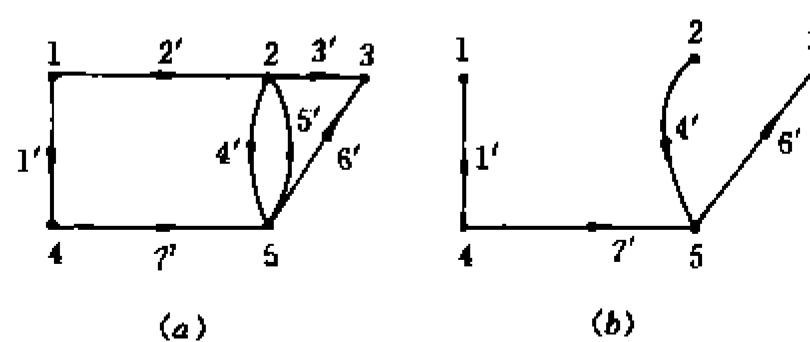


图 4 5-11

令：

$$V_C = \begin{pmatrix} v_2 \\ v_3 \\ v_5 \end{pmatrix}, \quad V_T = \begin{pmatrix} v_1 \\ v_4 \\ v_6 \\ v_7 \end{pmatrix} = \begin{pmatrix} e(t) \\ v_4 \\ v_6 \\ v_7 \end{pmatrix},$$

$$I_C = \begin{pmatrix} i_2 \\ i_3 \\ i_5 \end{pmatrix}, \quad I_T = \begin{pmatrix} i \\ i_4 \\ i_6 \\ i_7 \end{pmatrix}。$$

但

$$\begin{bmatrix} v_2 \\ v_3 \\ v_5 \end{bmatrix} = \begin{bmatrix} L_2 \dfrac{di_2}{dt} \\ Ri_3 \\ e_5(t) \end{bmatrix}, \qquad \begin{bmatrix} i_1 \\ i_4 \\ i_6 \\ i_7 \end{bmatrix} = \begin{bmatrix} i_1 \\ C_4 \dfrac{dv_4}{dt} \\ C_6 \dfrac{dv_6}{dt} \\ \dfrac{1}{R_7} v_7 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ \end{matrix} \begin{matrix} \begin{bmatrix} 1 & 0 & 0 & \vdots & 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & \vdots & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & \vdots & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & & -1 & 0 & 0 & 1 \end{bmatrix} \\ \begin{matrix} 2' & 3' & 5' & 1' & 4' & 6' & 7' \end{matrix} \end{matrix},$$

$$\boldsymbol{B}_{11} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad \boldsymbol{B}_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}。$$

$$\boldsymbol{B}_{12}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix},$$

$$\therefore \quad \boldsymbol{K} = \boldsymbol{B}_{11}^{T}(\boldsymbol{B}_{12}^{-1})^{T} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}。$$

故得：

$$\begin{bmatrix} L_2 \dfrac{di_2}{dt} \\ R_3 i_3 \\ e_5(t) \\ i_1 \\ C_4 \dfrac{dv_4}{dt} \\ C_6 \dfrac{dv_6}{dt} \\ G_7 v_7 \end{bmatrix} = \left[\begin{array}{ccc|cccc} 0 & 0 & 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} i_2 \\ i_3 \\ i_5 \\ e_1 \\ v_4 \\ v_6 \\ v_7 \end{bmatrix}。$$

展开得：

$$\begin{cases} L_2 \dfrac{di_2}{dt} = \quad v_4 + v_7, \\ C_4 \dfrac{dv_4}{dt} = i_2 - i_3 + i_5, \\ C_6 \dfrac{dv_6}{dt} = - i_3, \\ R_3 i_3 = v_4 + v_6, \\ G_7 v_7 = - i_2. \end{cases}$$

$\because \quad G_7 = \dfrac{1}{R_7} \quad \therefore \quad v_7 = \quad R_7 i_2$。

所以

$$\begin{cases} L_2 \dfrac{di_2}{dt} = \quad v_4 - R_7 i_2, \\ C_4 \dfrac{dv_4}{dt} = i_2 - \dfrac{1}{R_3}(v_4 + v_6) + i_5, \\ C_6 \dfrac{dv_6}{dt} = \quad \dfrac{1}{R_3}(v_4 + v_6). \end{cases}$$

或写成矩阵形式得关于 i_2、v_4、v_6 的微分方程组如下：

$$\frac{d}{dt}\begin{pmatrix} i_2 \\ v_4 \\ v_6 \end{pmatrix} = \begin{pmatrix} -\dfrac{R_7}{L_2} & -\dfrac{1}{L_2} & 0 \\ \dfrac{1}{C_4} & \dfrac{-1}{C_4R_3} & \dfrac{1}{C_4R_4} \\ 0 & \dfrac{-1}{C_6R_3} & \dfrac{-1}{C_6R_3} \end{pmatrix}\begin{pmatrix} i_2 \\ v_4 \\ v_6 \end{pmatrix} + \begin{pmatrix} \dfrac{1}{L_2} & 0 \\ 0 & \dfrac{1}{C_4} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} e_1(t) \\ i_5 \end{pmatrix}。$$

若令

$$\boldsymbol{X} = \begin{pmatrix} i_2 \\ v_4 \\ v_6 \end{pmatrix}, \qquad \boldsymbol{M} = \begin{pmatrix} \dfrac{-R_7}{L_2} & -\dfrac{1}{L_2} & 0 \\ \dfrac{1}{C_4} & \dfrac{-1}{C_4R_3} & \dfrac{-1}{C_4R_4} \\ 0 & \dfrac{-1}{C_6R_6} & \dfrac{-1}{C_6R_6} \end{pmatrix},$$

$$\boldsymbol{N} = \begin{pmatrix} \dfrac{1}{L_2} & 0 \\ 0 & \dfrac{1}{C_4} \\ 0 & 0 \end{pmatrix} \qquad \boldsymbol{U} = \begin{pmatrix} e_1(t) \\ i_5(t) \end{pmatrix},$$

则得：

$$\frac{d\boldsymbol{X}}{dt} = \boldsymbol{MX} + \boldsymbol{NU}。$$

§6　若干特殊情形

上面介绍了状态变量法，其中很主要的一个步骤在于找到一棵所谓“正常树”，使得树枝容下全部的电容和电压源。如若电容的边过多，超过树枝所能容下的范围，或电感过多，任何余树都容不下，这时如何处理？下面举例说明。

例 1：

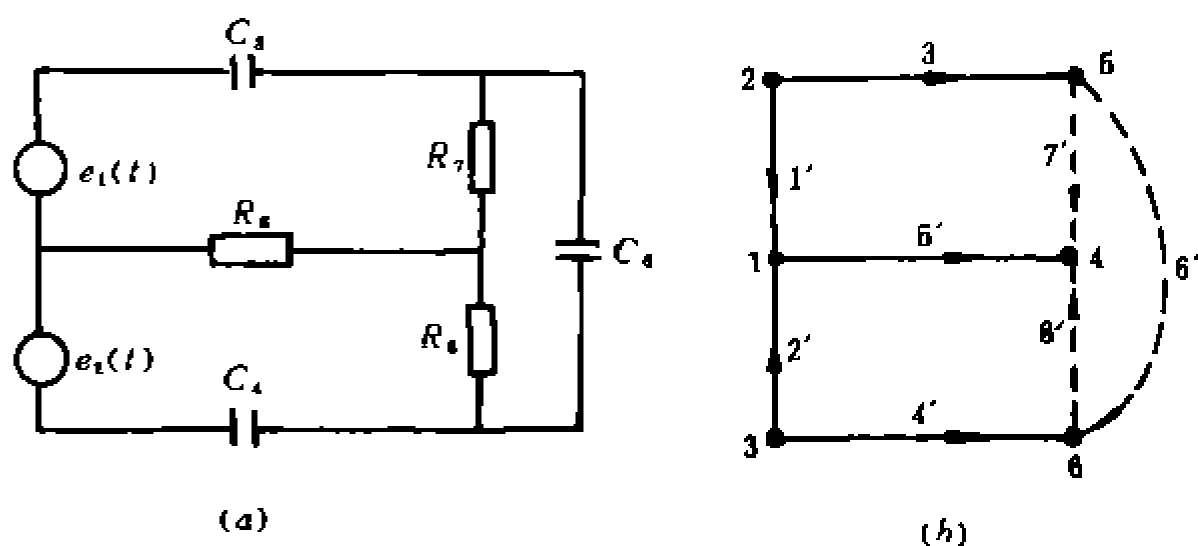


图 4-6-1

图 4-6-1 的图(b)是(a)中电路的树(实线)和余树(虚线)。

$$
B=\begin{array}{c}1\\2\\3\\4\\5\\6\\ \end{array}\left[\begin{array}{ccc:ccccc}
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0\\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 1\\
1 & 1 & 0 & 0 & 0 & -1 & 0 & 0\\
1 & 0 & 1 & 0 & 0 & 0 & -1 & 0
\end{array}\right],
$$
$$
\quad\;\; 6' \quad 7' \quad 8' \quad 1' \quad 2' \quad 3' \quad 4' \quad 5'
$$

$$
B_1=\begin{array}{c}2\\3\\4\\5\\6\\ \end{array}\left[\begin{array}{cccccccc}
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0\\
0 & -1 & -1 & 0 & 0 & 0 & 0 & 1\\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0\\
1 & 0 & 1 & 0 & 0 & 0 & -1 & 0
\end{array}\right]。
$$
$$
\quad\;\; 6' \quad 7' \quad 8' \quad 1' \quad 2' \quad 3' \quad 4' \quad 5'
$$

$$
B_{12}=\begin{bmatrix}
1 & 0 & 1 & 0 & 0\\
0 & 1 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 1\\
0 & 0 & -1 & 0 & 0\\
0 & 0 & 0 & -1 & 0
\end{bmatrix},\qquad
B_{12}'=\begin{bmatrix}
1 & 0 & 0 & 1 & 0\\
0 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & -1\\
0 & 0 & -1 & 0 & 0
\end{bmatrix}。
$$

$$\therefore\quad \boldsymbol{K}=\boldsymbol{B}_{11}^{T}(\boldsymbol{B}_{12}^{-1})^{T}=\begin{pmatrix}0&0&0&1&-1\\0&0&-1&1&0\\0&0&1&0&1\end{pmatrix}\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&0&0&-1\\1&0&-1&0&0\\0&1&0&1&0\end{pmatrix}$$

$$=\begin{pmatrix}1&1&-1&1&0\\1&0&-1&0&1\\0&1&0&-1&1\end{pmatrix}。$$

本例与前不同，电容的边过多以至不存在一棵树使得它的树枝包含了所有的电容。

$$\boldsymbol{V}_C=\begin{pmatrix}v_6\\v_7\\v_8\end{pmatrix}=\begin{pmatrix}v_6\\R_7i_7\\R_8i_8\end{pmatrix},\qquad \boldsymbol{I}_C=\begin{pmatrix}C_6\dfrac{dv_6}{dt}\\i_7\\i_8\end{pmatrix},$$

$$\boldsymbol{I}_T=\begin{pmatrix}i_1\\i_2\\i_3\\i_4\\i_5\end{pmatrix}=\begin{pmatrix}i_1\\i_2\\C_3\dfrac{dv_3}{dt}\\C_4\dfrac{dv_4}{dt}\\G_5v_5\end{pmatrix},\qquad \boldsymbol{V}_T=\begin{pmatrix}e_1(t)\\e_2(t)\\v_3\\v_4\\v_5\end{pmatrix}。$$

以之代入公式：

$$\begin{pmatrix}\boldsymbol{V}_C\\\boldsymbol{I}_T\end{pmatrix}=\begin{pmatrix}\boldsymbol{O}&\boldsymbol{K}\\-\boldsymbol{K}^T&\boldsymbol{O}\end{pmatrix}\begin{pmatrix}\boldsymbol{I}_C\\\boldsymbol{V}_T\end{pmatrix},$$

得：

$$\begin{pmatrix}v_6\\R_7i_7\\R_8i_8\\i_1\\i_2\\C_3\dfrac{dv_3}{dt}\\C_4\dfrac{dv_4}{dt}\\G_5v_5\end{pmatrix}=\begin{pmatrix}0&0&0&1&1&1&1&0\\0&0&0&1&0&1&0&1\\0&0&0&0&1&0&1&1\\-1&1&0&0&0&0&0&0\\1&0&-1&0&0&0&0&0\\1&1&0&0&0&0&0&0\\1&0&1&0&0&0&0&0\\0&-1&-1&0&0&0&0&0\end{pmatrix}\begin{pmatrix}C_6\dfrac{dv_6}{dt}\\i_7\\i_8\\e_1(t)\\e_2(t)\\v_3\\v_4\\v_5\end{pmatrix}。$$

展开得：

$$C_3\frac{dv_3}{dt}=C_6\frac{dv_5}{dt}+i_7,\tag{1}$$

$$C_4\frac{dv_4}{dt}=C_6\frac{dv_6}{dt}+i_8,\tag{2}$$

但
$$v_6=e_1(t)-e_2(t)-v_3+v_4,\tag{3}$$
对(3)求导得：
$$\frac{dv_6}{dt}=e_1'(t)-e_2'(t)-\frac{dv_3}{dt}+\frac{dv_4}{dt}。\tag{4}$$
又
$$i_7=[e_1(t)-v_3+v_5]/R_7,\tag{5}$$
$$i_8=[e_2(t)-v_4+v_5]/R_8,\tag{6}$$
$$v_5=-R_5(i_7+i_8)。\tag{7}$$
将(7)代入(5)、(6)得：
$$i_7=[e_1(t)-v_3-R_5i_7-R_5i_8]/R_7,$$
$$i_8=[e_2(t)-v_4-R_5i_7-R_5i_8]/R_8。$$
整理得关于 i_7、i_8 的联立方程组：
$$\begin{cases}\left(1+\dfrac{R_5}{R_7}\right)i_7+\dfrac{R_5}{R_7}i_8=\dfrac{1}{R_7}e_1(t)-\dfrac{1}{R_7}v_3, & (8)\\ \dfrac{R_5}{R_8}i_7+\left(1+\dfrac{R_5}{R_8}\right)i_8=\dfrac{1}{R_8}e_2(t)-\dfrac{1}{R_8}v_4。 & (9)\end{cases}$$
利用Cramer 法则解这方程组得：
$$i_7=\begin{vmatrix}\dfrac{1}{R_7}e_1(t)-\dfrac{1}{R_7}v_3 & \dfrac{R_5}{R_7}\\ \dfrac{1}{R_8}e_2(t)-\dfrac{1}{R_8}v_4 & 1+\dfrac{R_5}{R_8}\end{vmatrix}\div\begin{vmatrix}1+\dfrac{R_5}{R_7} & \dfrac{R_5}{R_7}\\ \dfrac{R_5}{R_8} & 1+\dfrac{R_5}{R_8}\end{vmatrix}$$
$$=\frac{-\left(1+\dfrac{R_5}{R_8}\right)\dfrac{v_3}{R_7}+\dfrac{R_5}{R_7R_8}v_4+\left(1+\dfrac{R_5}{R_8}\right)\dfrac{e_1(t)}{R_7}-\dfrac{R_5}{R_7R_8}e_2(t)}{1+R_5\left(\dfrac{1}{R_7}+\dfrac{1}{R_8}\right)}$$
$$=\frac{-(R_8+R_8)v_3+R_5v_4+(R_8+R_5)e_1(t)-R_5e_2(t)}{R_7R_8+R_5R_7+R_5R_8}。\tag{10}$$
$$i_8=\begin{vmatrix}1+\dfrac{R_5}{R_7} & \dfrac{1}{R_7}e_1(t)-\dfrac{1}{R_7}v_3\\ \dfrac{R_5}{R_8} & \dfrac{1}{R_8}e_2(t)-\dfrac{1}{R_8}v_4\end{vmatrix}\div\begin{vmatrix}1+\dfrac{R_5}{R_7} & \dfrac{R_5}{R_7}\\ \dfrac{R_5}{R_8} & 1+\dfrac{R_5}{R_8}\end{vmatrix}$$
$$=\frac{\dfrac{R_5v_3}{R_7R_8}-\left(\dfrac{1}{R_8}+\dfrac{R_5}{R_7R_8}\right)v_4+\left(\dfrac{1}{R_8}+\dfrac{R_5}{R_7R_8}\right)e_2(t)-\dfrac{R_5}{R_7R_8}e_1(t)}{1+R_5\left(\dfrac{1}{R_7}+\dfrac{1}{R_8}\right)}$$
$$=\frac{R_5v_3-(R_7+R_5)v_4+(R_7+R_5)e_2(t)-R_5e_1(t)}{R_7R_8+R_5R_7+R_8R_5}。\tag{11}$$
把(10)(11)代入(1)、(2)分别得：
$$C_3\frac{dv_3}{dt}=C_6\left[e_1'(t)-e_2(t)-\frac{dv_3}{dt}+\frac{dv_4}{dt}\right]$$
$$+\frac{-(R_8+R_5)v_3+R_5v_4+(R_8+R_5)e_1(t)-R_5e_2(t)}{R_7R_8+R_5R_7+R_5R_8}。$$

$$\therefore \qquad (C_3+C_6)\frac{dv_3}{dt}-C_6\frac{dv_4}{dt}$$

$$=\frac{-(R_5+R_8)(v_3-e_1(t))+R_5(v_4-e_2(t))}{R_7R_8+R_5R_7+R_5R_8}+C_6(e_1'(t)-e_2'(t))。\qquad (12)$$

$$C_4\frac{dv_4}{dt}=-C_6\left[e_1'(t)-e_2'(t)-\frac{dv_3}{dt}+\frac{dv_4}{dt}\right]$$

$$+\frac{R_5v_3-(R_7+R_5)v_4+(R_7+R_5)e_2(t)-R_5e_1(t)}{R_5R_7+R_5R_8+R_7R_8}。$$

$$\therefore \qquad -C_6\frac{dv_3}{dt}+(C_4+C_6)\frac{dv_4}{dt}$$

$$=\frac{R_5(v_3-e_1(t))-(R_7+R_5)(v_4-e_2(t))}{R_5R_7+R_5R_8+R_7R_8}+C_6(e_2'(t)-e_1'(t))。\qquad (13)$$

把(12)(13)写成矩阵形式：

$$\begin{pmatrix} C_3+C_6 & -C_6 \\ -C_6 & C_4+C_6 \end{pmatrix}\begin{pmatrix} \frac{dv_3}{dt} \\ \frac{dv_4}{dt} \end{pmatrix}=\frac{1}{R_5R_7+R_5R_8+R_7R_8}$$

$$\begin{pmatrix} -R_5-R_8 & R_5 \\ R_5 & -R_5-R_7 \end{pmatrix}\left\{\begin{pmatrix} v_3 \\ v_4 \end{pmatrix}-\begin{pmatrix} e_1(t) \\ e_2(t) \end{pmatrix}\right\}+C_6\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\begin{pmatrix} e_1'(t) \\ e_2'(t) \end{pmatrix}。\qquad (14)$$

若令

$$A=\frac{1}{R_5R_7+R_5R_8+R_7R_8}\begin{pmatrix} C_3+C_6 & -C_6 \\ -C_6 & C_4+C_6 \end{pmatrix}^{-1}\begin{pmatrix} -R_5-R_8 & R_5 \\ R_5 & -R_5-R_7 \end{pmatrix},$$

$$B=C_6\begin{pmatrix} C_3+C_6 & -C_6 \\ -C_6 & C_4+C_6 \end{pmatrix}^{-1}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$

$$X=\begin{pmatrix} v_3 \\ v_4 \end{pmatrix}，\quad U=B\begin{pmatrix} e_1'(t) \\ e_2(t) \end{pmatrix}-A\begin{pmatrix} e_1(t) \\ e_2(t) \end{pmatrix},$$

则(14)可写成一阶常微分方程组：

$$\frac{dX}{dt}=AX+U。$$

例 2：

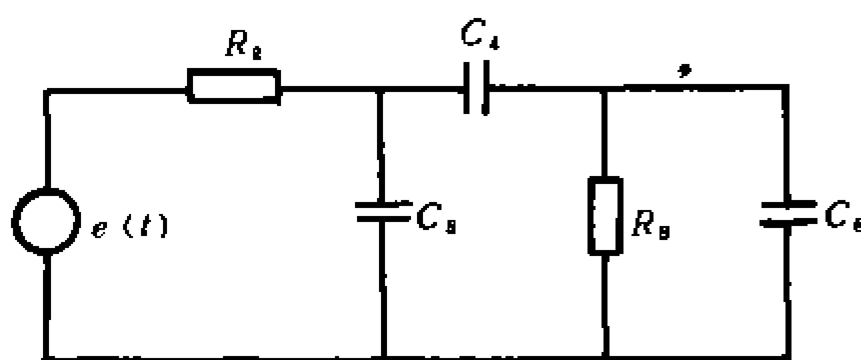


图 4-6-2

如图 4-6-2 所示的电路网络，其对应的图 G 和树 T 如图 4-6-3。

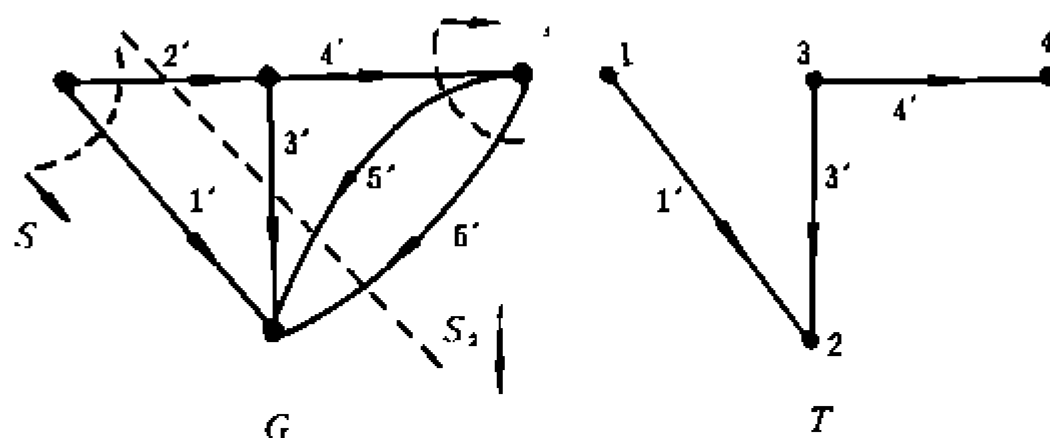


图 4-6-3

显而易见不存在一棵正常树，不过可在图 4-6-2 的带 * 号的边上加极微小电阻 r，使问题变为正常情况，从而得一近似解。详细的讨论留给读者作为练习。

$$
S_f = \begin{matrix} S_1 \\ S_2 \\ S_3 \end{matrix}\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 0 & 1 \end{bmatrix},
$$
$$
\qquad\qquad\quad 2' \quad 5' \quad 6' \quad 1' \quad 3' \quad 4'
$$

$$
K = S_{11}^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix}。
$$

$$
\begin{bmatrix} v_2 \\ v_5 \\ v_6 \\ i_1 \\ i_3 \\ i_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_2 \\ i_5 \\ i_6 \\ v_1 \\ v_3 \\ v_4 \end{bmatrix}。
$$

$$
\begin{bmatrix} v_2 \\ v_5 \\ v_6 \\ i \\ C_3 \dfrac{dv_3}{dt} \\ C_4 \dfrac{dv_4}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} G_2 v_2 \\ G_5 v_5 \\ C_6 \dfrac{dv_6}{dt} \\ e_1(t) \\ v_3 \\ v_4 \end{bmatrix}。
$$

这是电容过剩的情形。

$$
i_3 = C_3 \frac{dv_3}{dt},\ i_4 = C_4 \frac{dv_4}{dt},\ i_6 = C_6 \frac{dv_6}{dt},
$$

其中：

$$G_2=\frac{1}{R_2},\qquad G_5=\frac{1}{R_5},$$

$$C_3\frac{dv_3}{dt}=G_2v_2-G_5v_5-C_6\frac{dv_6}{dt},$$

$$C_4\frac{dv_4}{dt}=G_5v_5+C_6\frac{dv_6}{dt},$$

$$\begin{pmatrix}C_3\dfrac{dv_3}{dt}\\ C_4\dfrac{dv_4}{dt}\end{pmatrix}=\begin{pmatrix}G_2&-G_5\\0&G_5\end{pmatrix}\begin{pmatrix}v_2\\v_5\end{pmatrix}+\begin{pmatrix}-C_6\\C_6\end{pmatrix}\frac{dv_6}{dt}。$$

$$\therefore\qquad \begin{pmatrix}C_3&0\\0&C_4\end{pmatrix}\begin{pmatrix}\dfrac{dv_3}{dt}\\ \dfrac{dv_4}{dt}\end{pmatrix}=\begin{pmatrix}1&-1\\0&1\end{pmatrix}\begin{pmatrix}G_2&0\\0&G_5\end{pmatrix}\begin{pmatrix}v_2\\v_5\end{pmatrix}+\begin{pmatrix}-1\\1\end{pmatrix}C_6\frac{dv_6}{dt},$$

但

$$\begin{pmatrix}v_2\\v_5\end{pmatrix}=\begin{pmatrix}-1&0\\1&-1\end{pmatrix}\begin{pmatrix}v_3\\v_4\end{pmatrix}+\begin{pmatrix}1\\0\end{pmatrix}e_1(t),\qquad v_6=v_3-v_4,$$

$$\therefore\qquad \frac{dv_6}{dt}=\frac{dv_3}{dt}-\frac{dv_4}{dt}=(1\quad -1)\begin{pmatrix}\dfrac{dv_3}{dt}\\ \dfrac{dv_4}{dt}\end{pmatrix}。$$

$$\frac{d}{dt}\begin{pmatrix}v_3\\v_4\end{pmatrix}=\begin{pmatrix}1/C_3&0\\0&1/C_4\end{pmatrix}\left\{\begin{pmatrix}1&-1\\0&1\end{pmatrix}\begin{pmatrix}G_2&0\\0&G_5\end{pmatrix}\left[\begin{pmatrix}-1&0\\1&-1\end{pmatrix}\begin{pmatrix}v_3\\v_4\end{pmatrix}+\begin{pmatrix}1\\0\end{pmatrix}e(t)\right]\right.$$

$$\left.+\begin{pmatrix}-1\\1\end{pmatrix}(C_6\quad -C_6)\begin{pmatrix}\dfrac{dv_3}{dt}\\ \dfrac{dv_4}{dt}\end{pmatrix}\right\},$$

由于

$$\begin{pmatrix}-1\\1\end{pmatrix}(C_6\quad -C_6)=\begin{pmatrix}-C_6&C_6\\C_6&-C_6\end{pmatrix},$$

有

$$\left[\begin{pmatrix}1&0\\0&1\end{pmatrix}+\begin{pmatrix}C_6/C_3&-C_6/C_3\\-C_6/C_4&C_6/C_4\end{pmatrix}\right]\frac{d}{dt}\begin{pmatrix}v_3\\v_4\end{pmatrix}$$

$$=\begin{pmatrix}\dfrac{1}{C_3}&-\dfrac{1}{C_3}\\0&\dfrac{1}{C_4}\end{pmatrix}\begin{pmatrix}-G_2&0\\G_5&-G_5\end{pmatrix}\begin{pmatrix}v_3\\v_4\end{pmatrix}+\begin{pmatrix}\dfrac{1}{C_3}&-\dfrac{1}{C_3}\\0&\dfrac{1}{C_4}\end{pmatrix}\begin{pmatrix}G_2\\0\end{pmatrix}e(t)。$$

即：

$$\begin{pmatrix}1+\dfrac{C_6}{C_3}&-\dfrac{C_6}{C_3}\\-\dfrac{C_6}{C_4}&1+\dfrac{C_6}{C_4}\end{pmatrix}\frac{d}{dt}\begin{pmatrix}v_3\\v_4\end{pmatrix}=\begin{pmatrix}\dfrac{-1}{C_3R_2}-\dfrac{1}{C_3R_5}&\dfrac{1}{C_3R_5}\\ \dfrac{+1}{C_4R_5}&\dfrac{-1}{C_4R_5}\end{pmatrix}\begin{pmatrix}v_3\\v_4\end{pmatrix}+\begin{pmatrix}\dfrac{1}{C_3R_2}\\0\end{pmatrix}e(t)。$$

$$\frac{d}{dt}\begin{pmatrix} v_3 \\ v_4 \end{pmatrix} = M\begin{pmatrix} v_3 \\ v_4 \end{pmatrix} + NU$$

其中

$$M = \begin{pmatrix} 1+\dfrac{C_6}{C_3} & -\dfrac{C_6}{C_3} \\ \dfrac{C_6}{C_4} & 1+\dfrac{C_6}{C_4} \end{pmatrix} \begin{pmatrix} \dfrac{-1}{C_3C_5}-\dfrac{1}{C_3R_2} & \dfrac{1}{C_3R_5} \\ \dfrac{1}{C_4R_5} & \dfrac{-1}{C_4R_5} \end{pmatrix},$$

$$N = \begin{pmatrix} 1+\dfrac{C_6}{C_3} & -\dfrac{C_6}{C_3} \\ \dfrac{C_6}{C_4} & 1+\dfrac{C_6}{C_4} \end{pmatrix} \begin{pmatrix} \dfrac{1}{R_2C_3} \\ 0 \end{pmatrix},$$

$U = e_s(t)$。

例 3：

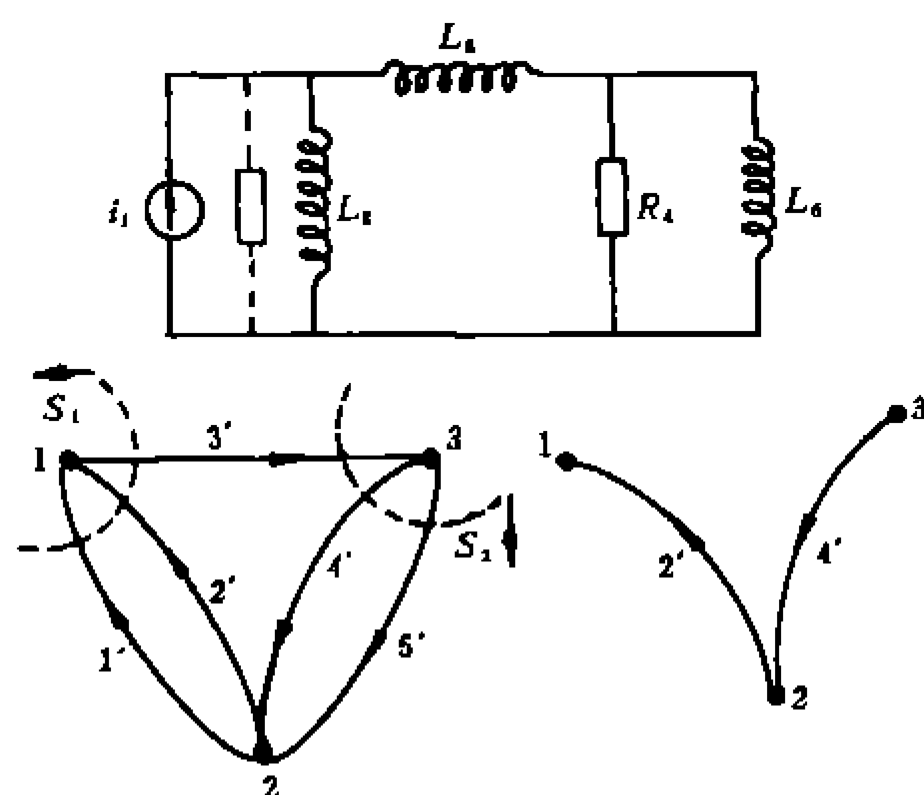


图 4 6-4

$$S_f = \begin{pmatrix} 1 & -1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} S_1 \\ S_2, \end{matrix}$$
$$\quad 1' \quad 3' \quad 5' \quad 2' \quad 4'$$

可得：

$$\begin{pmatrix} v_1 \\ v_3 \\ v_5 \\ i_2 \\ i_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_1 \\ i_3 \\ i_5 \\ v_2 \\ v_4 \end{pmatrix},$$

其中

$$
K = S_{11}^T = \begin{bmatrix} 1 & 0 \\ -1 & -1 \\ 0 & 1 \end{bmatrix}。
$$

这是电感太多的例子。可在图 4-6-4 的虚线边加一极大的电阻，使问题变为正常情形，从而得一近似解。

现在树枝上有：

$$
v_2 = L_2 \frac{di_2}{dt},
$$

在余树枝上有：

$$
v_3 = L_3 \frac{di_3}{dt}, \qquad v_5 = L_5 \frac{di_5}{dt},
$$

以之代人上式得到

$$
\begin{bmatrix} v_1 \\ L_3 \dfrac{di_3}{dt} \\ L_5 \dfrac{di_5}{dt} \\ i_2 \\ i_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_1(t) \\ i_3 \\ i_5 \\ L_2 \dfrac{di_2}{dt} \\ R_4 i_4 \end{bmatrix},
$$

所以

$$
\begin{bmatrix} L_3 \dfrac{di_3}{dt} \\ L_5 \dfrac{di_5}{dt} \end{bmatrix} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \begin{bmatrix} L_2 \dfrac{di_2}{dt} \\ R_4 i_4 \end{bmatrix}。
$$

但

$$
\begin{aligned}
& i_2 = -i_1(t) + i_3, \\
& i_4 = i_3 - i_5, \\
& L_3 \frac{di_3}{dt} = -L_2 \frac{di_2}{dt} - R_4 i_4,
\end{aligned}
$$

所以

$$
L_3 \frac{di_3}{dt} = -L_2\left[\frac{di_3}{dt} - \frac{di_1}{dt}\right] - R_4(i_3 - i_5)。
$$

即

$$
(L_3 + L_2)\frac{di_3}{dt} = -R_4 i_3 + R_4 i_5 + L_2 \frac{di_1}{dt}。
$$

∴

$$
\frac{d}{dt}\begin{pmatrix} i_3 \\ i_5 \end{pmatrix} = \begin{bmatrix} \dfrac{-R_4}{L_2+L_3} & \dfrac{R_4}{L_2+L_3} \\ \dfrac{R_4}{L_5} & \dfrac{-R_4}{L_5} \end{bmatrix} \begin{pmatrix} i_3 \\ i_6 \end{pmatrix} + \begin{bmatrix} \dfrac{L_2}{L_2+L_3} \\ 0 \end{bmatrix} \frac{di_1}{dt}
$$

令

$$
X = \begin{pmatrix} i_3 \\ i_5 \end{pmatrix} - \begin{bmatrix} \dfrac{L_2 i_1}{L_2 + L_3} \\ 0 \end{bmatrix},
$$

则得：

$$\frac{dX}{dt}=\begin{pmatrix}\frac{-R_4}{L_2+L_3} & \frac{R_4}{L_2+L_3}\\ \frac{R_4}{L_5} & \frac{-R_4}{L_5}\end{pmatrix}X+\begin{pmatrix}\frac{-R_4}{L_2+L_3} & \frac{R_4}{L_2+L_3}\\ \frac{R_4}{L_5} & \frac{-R_4}{L_5}\end{pmatrix}\begin{pmatrix}\frac{L_2}{L_2+L_3}i_1\\ 0\end{pmatrix}=MX+NU。$$

其中：

$$M=\begin{pmatrix}\frac{-R_4}{L_2+L_3} & \frac{R_4}{L_2+L_3}\\ \frac{R_4}{L_5} & \frac{-R_3}{L_5}\end{pmatrix},\quad N=\begin{pmatrix}\frac{-R_4L_2}{(L_2+L_3)^2} & \frac{R_4L_2}{(L_2+L_3)^2}\\ \frac{R_4L_2}{L_5(L_2+L_3)} & \frac{-R_4L_2}{L_5(L_2+L_3)}\end{pmatrix},$$

$U=i_1$。

习　题

1. 列出下面电路的状态方程。

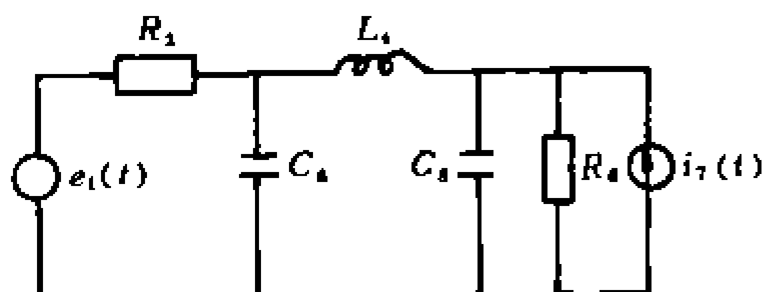


图　习题 1

2. 利用状态变量法解下面电路问题。

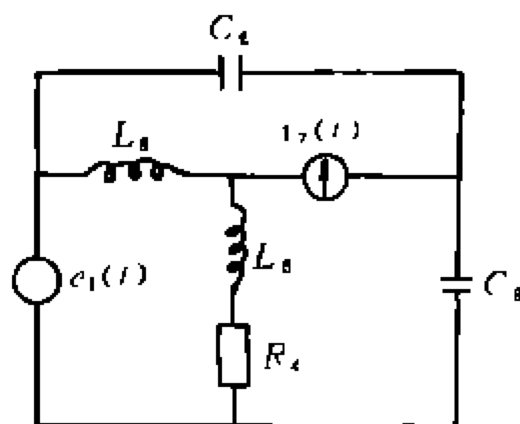


图　习题 2

3. 列出下列电路的状态方程。

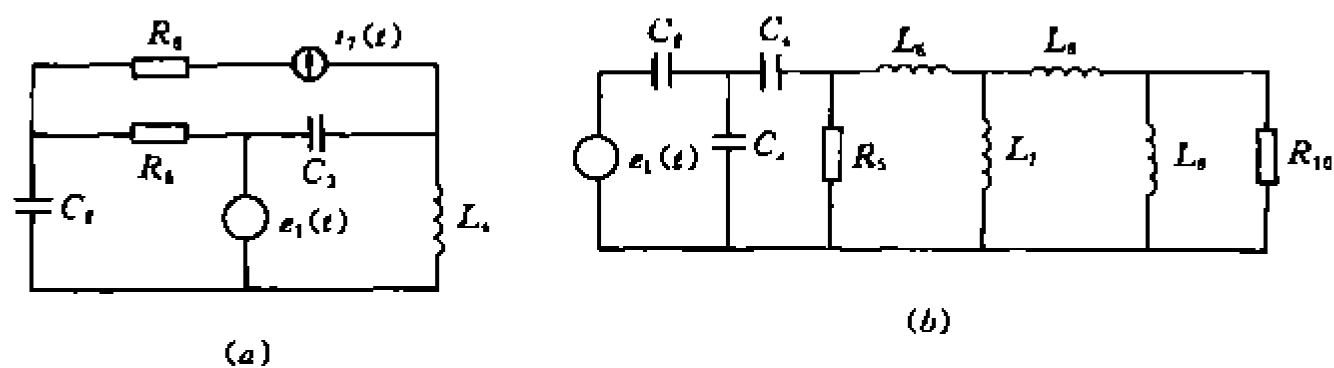


图　习题 3

4. 求下图电路的状态方程。

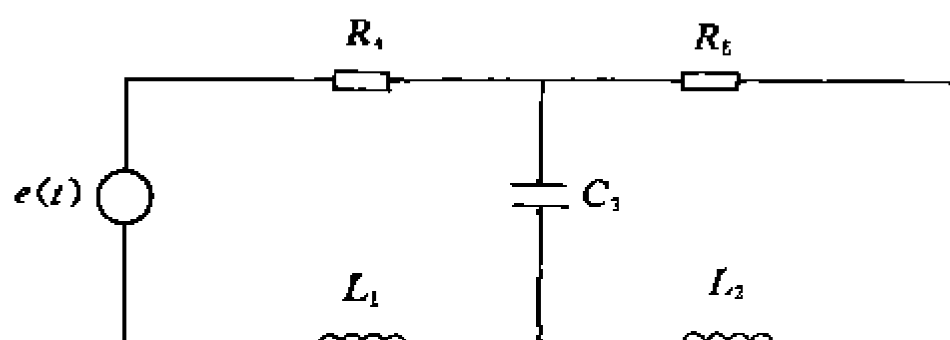


图　习题 4

5. 试列出下面网络的状态方程。

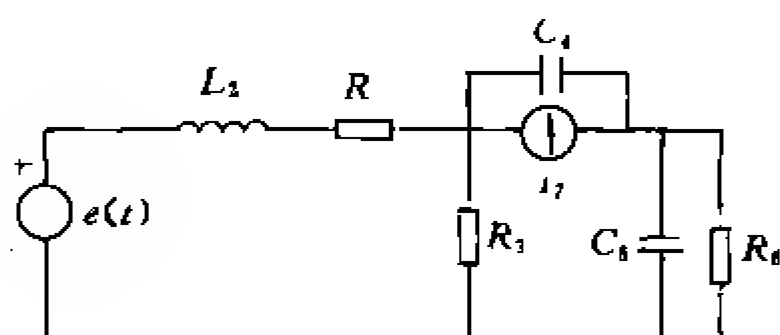


图　习题 5

6. 试列出下列网络的状态方程。

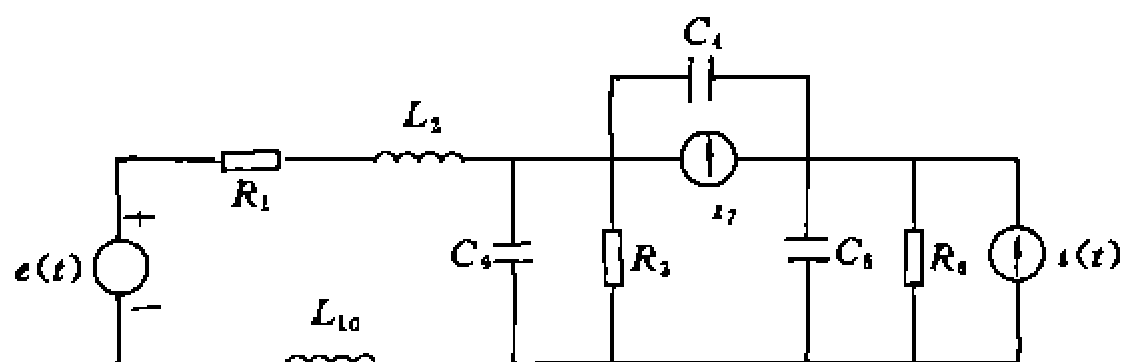


图　习题 6

第五章　信号流图问题

信号流图是一种顶点和弧都具有权的有向图，它是用图的结构表示线性方程组中诸变量间关系的一种方法。对于许多物理系统来说，它的输入和输出满足某代数方程组，可以根据它的特性直接作出信号流图，再由信号流图按一定规则可以写出对应方程组的解。从数学观点来讲，信号流图也是解方程组的一种方法，由于这种方法直观、灵活并使用简便，在工程系统，特别是线性系统（线性网络）中得到广泛的应用，成为分析线性系统的一种有力工具。

§1　矩阵与 Coates 流图

Coates 流图是应用较广泛的一种信号流图。这种图是把矩阵用一种有向图来表示，由于矩阵和线性方程组有对应关系，因此所得到的图可作为对应于方程组的信号流图。

矩阵

$$A=(a_{ij})_{n\times n}$$

可用一个 n 个顶点的图表示，即对于非零元素 a_{ij} 可以从 v_i 引一有向边到 v_j 点，并令它的权为 a_{ij}，如图 5-1-1 所示。

图　5-1-1

这样一 n 阶矩阵对应一 n 个顶点的带权的有向图，叫做 Coates 图。例如：

$$A=\begin{bmatrix}0&0&1&2\\0&0&2&1\\1&2&0&0\\2&4&0&0\end{bmatrix}$$

对应一有向图 5-1-2。

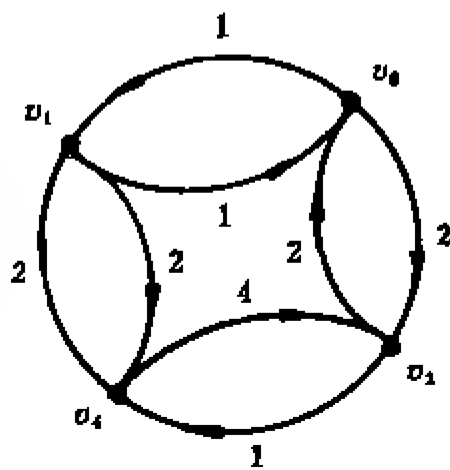


图　5-1-2

于是有些线性代数的问题便化为图论的问题了。比如一非强连通的图 G，它的邻接矩阵为 $A=(a_{ij})_{n\times n}$，则适当地编排顶点的次序使得顶点 $v_1,v_2,\cdots,v_r$ 形成一强连通块，则存在一置换矩阵 P 使得

$$PAP^T=\begin{bmatrix}A_{11}&O\\ \underbrace{A_{21}}_{r}&\underbrace{A_{22}}_{n-r}\end{bmatrix}\begin{matrix}\}r\\ \}n-r\end{matrix}$$

对于矩阵 $A=(a_{ij})_{n\times n}$，若存在 $n\times n$ 的置换矩阵 P 使得：

$$PAP^T=\begin{bmatrix}A_{11}&O\\ \underbrace{A_{21}}_{r}&\underbrace{A_{22}}_{n-r}\end{bmatrix}\begin{matrix}\}r\\ \}n-r\end{matrix}$$

时，则称矩阵 A 是可化约的，否则是不可化约的。

若 $A=(a_{ij})_{n\times n}$ 是可化约的，则方程组 $AX=B$ 可分解为：

$$\begin{pmatrix} A_{11} & O \\ A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} X_1 \\ X_2 \end{pmatrix}=\begin{pmatrix} B_1 \\ B_2 \end{pmatrix}。$$

即：

$$\begin{cases} A_{11}X_1=B_1, \\ A_{22}X_2=B_2-A_{21}X_1, \end{cases}$$

其中

$$X_1=\begin{pmatrix} x_1 \\ \vdots \\ x_r \end{pmatrix},\quad X_2=\begin{pmatrix} x_{r+1} \\ \vdots \\ x_n \end{pmatrix}。$$

不难证明：矩阵 $A=(a_{ij})_{n\times n}$ 为不可化约矩阵的充要条件是：它所对应的图为强连通的。

§2　代数方程组与 Mason 信号流图

代数方程组也可以化为相应的图。以下列的代数方程组为例：

$$\begin{cases} a_{11}x_1+a_{12}x_2+a_{13}x_3+a_{14}x_4=b_1, \\ a_{21}x_1+a_{22}x_2+a_{23}x_3+a_{24}x_4=b_2, \\ a_{31}x_1+a_{32}x_2+a_{33}x_3+a_{34}x_4=b_3, \\ a_{41}x_1+a_{42}x_2+a_{43}x_3+a_{44}x_4=b_4。 \end{cases} \tag{1}$$

先把方程组化成下面形式：

$$\begin{cases} x_1=b_1'+a_{11}'x_1+a_{12}'x_2+a_{13}'x_3+a_{14}'x_4, \\ x_2=b_2'+a_{21}'x_1+a_{22}'x_2+a_{23}'x_3+a_{24}'x_4, \\ x_3=b_3'+a_{31}'x_1+a_{32}'x_2+a_{33}'x_3+a_{34}'x_4, \\ x_4=b_4'+a_{41}'x_1+a_{42}'x_2+a_{43}'x_3+a_{44}'x_4。 \end{cases} \tag{2}$$

这方程组可用图来表示它，表示的法则为：

(a) 用

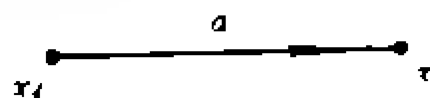


图　5-2-1

表示 $x_j=ax_i$。

(b) 用

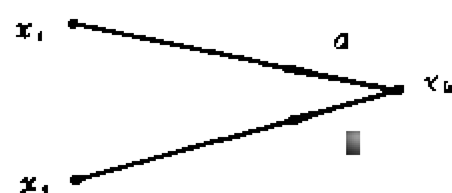


图　5-2-2

表示 $x_k = ax_i + bx_j$。则(2)可用图5-2-3表示。

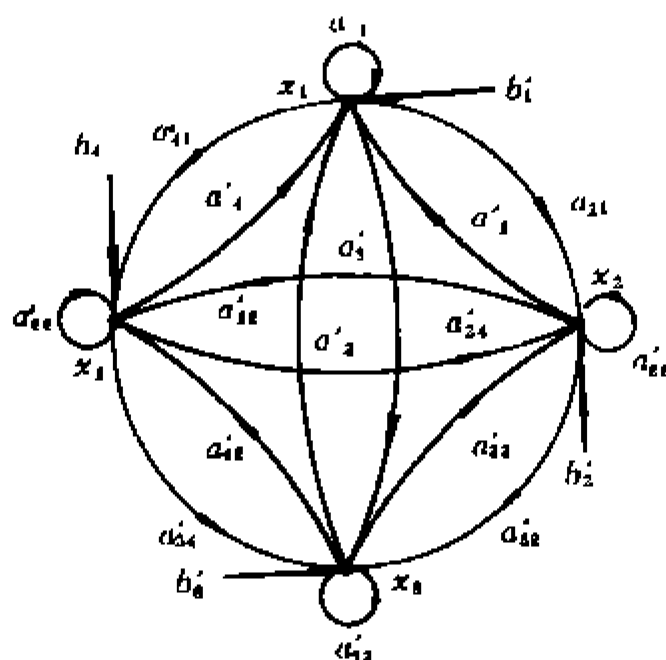


图 5-2-3

这样的图叫做方程组(1)的 Mason 信号流图。

例1：

$$\begin{cases} x_1 \quad x_2 + x_3 - 0, \\ 2x_1 + 2x_2 + x_3 \quad 0, \\ \quad x_1 \quad x_2 + x_3 = u. \end{cases}$$

把这方程组化为：

$$\begin{cases} x_1 - x_2 \quad x_3, \\ x_2 - 2x_1 + 3x_2 + x_1, \\ x_3 - x_1 + x_2 + u. \end{cases}$$

这方程组可用图5-2-4表示。

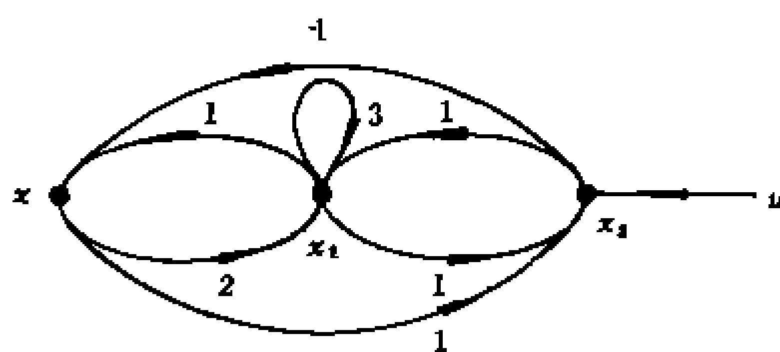


图 5-2-4

§3 信号流图的运算

后面我们将讨论如何通过信号流图求解方程组的问题，这个问题涉及到流图的简化规则这一节介绍几个方程组的图的运算法。通过例子加以说明，要求准确地掌握图的运算规律。

（1）加法规则（并联法则）

两个顶点间 m 个同向边可以用一条与它们同向的边代替，该边的增益为 m 个同向边的权的和（见图5-3-1）。

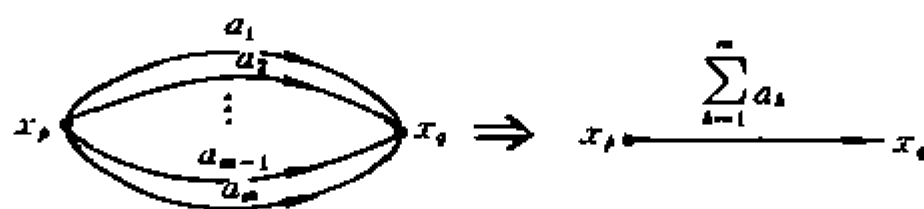


图 5-3-1

（2）乘法法则（串联法则）

$m+1$ 个顶点 $x_1, x_2, \cdots, x_m, x_{m+1}$，若有边 (x_1, x_2)，(x_2, x_3)，$\cdots$，(x_{m-1}, x_m)，(x_m, x_{m+1}) 相连，则可用有向边 (x_1, x_{m+1}) 代替，(x_1, x_{m+1}) 的增益为有向边 (x_1, x_2)，$\cdots$，(x_{m-1}, x_m)，(x_m, x_{m+1}) 的权的乘积（见图5-3-2）。

图 5-3-2

特别当 $m=2$ 时有：

图 5-3-3

（3）顶点消去法则

先看一个特殊情况：

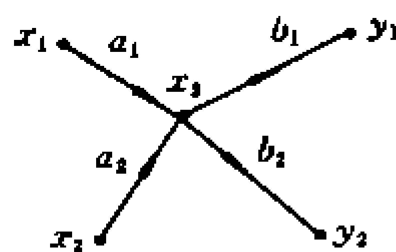


图 5-3-4

$$x_3=a_1x_1+a_2x_2,$$
$$y_1=b_1x_3=a_1b_1x_1+a_2b_1x_2,$$
$$y_2=b_2x_3=a_1b_2x_1+a_2b_2x_2。$$

故有

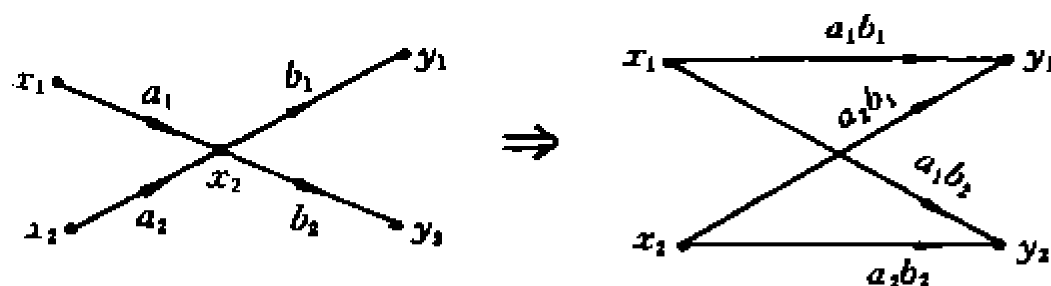


图 5-3-5

一般地对于以 x 为终点的有向边(x_1,x),(x_2,x),…,(x_m,x)和以 x 为始点的有向边(x,y_1),(x,y_2),…,(x,y_n),其增益分别是 $a_1,a_2,\cdots,a_m$ 和 $b_1,b_2,\cdots,b_n$。则可将顶点 x 消去,得有向边:

$$(x_1,y_1),\ (x_2,y_1),\cdots\cdots,(x_m,y_1),$$
$$(x_1,y_2),\ (x_2,y_2),\cdots\cdots,(x_m,y_2),$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$
$$(x_1,y_n),\ (x_2,y_n),\cdots\cdots,(x_m,y_n)。$$

其增益分别为:

$$a_1b_1,\ a_2b_1,\ \cdots\cdots,\ a_mb_1,$$
$$a_1b_2,\ a_2b_2,\ \cdots\cdots,\ a_mb_2,$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$
$$a_1b_n,\ a_2b_n,\cdots\cdots,a_mb_n。$$

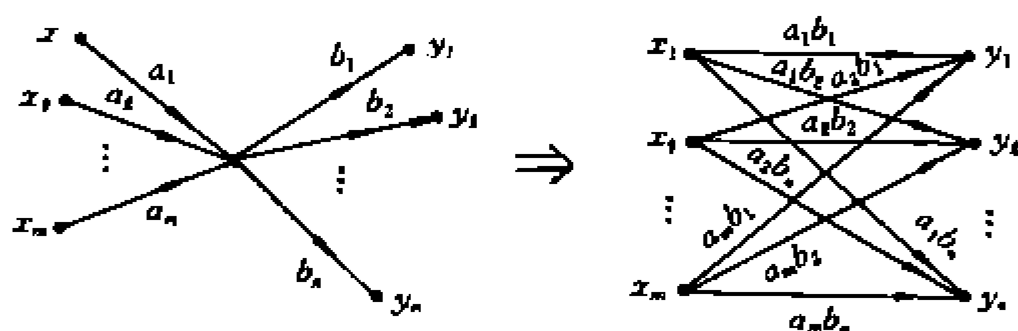


图 5 3 6

(4) 自环消去法则

先看几个简单的例子:

$$x_2=a_1x_1+a_2x_2$$

$$(1-a_2)x_2=a_1x_1 \qquad \therefore\ x_2=\frac{a_1}{1-a_2}x_1。$$

故得(图5 3 7):

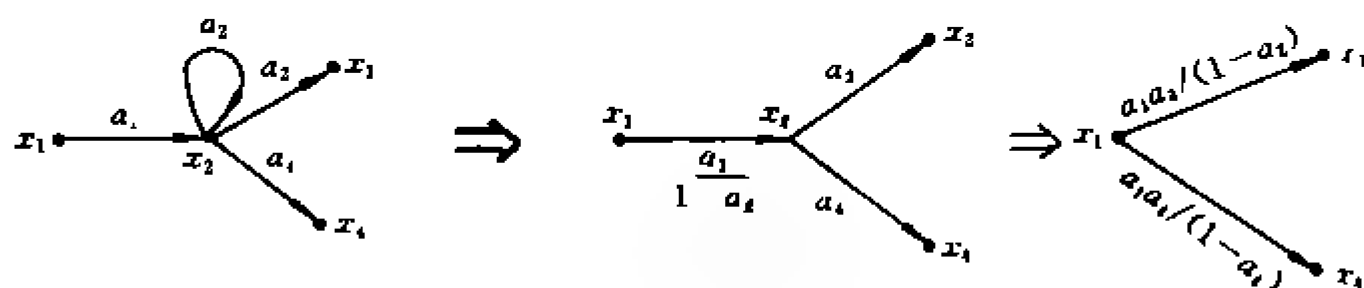


图 5 3 7

同样如(图5-3-8)。

一般地对于有 m 条以 x 为终点的有向边和自环的顶点 x,如将 m 条有向边上的增益除以(1 环的增益),则可将自环消去,以 x 为起点的有向边的增益不变。(见图5-3-9)。

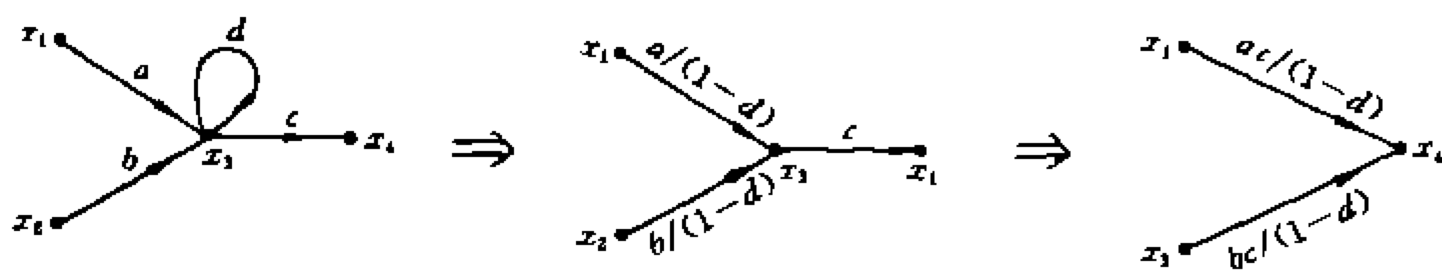


图 5-3-8

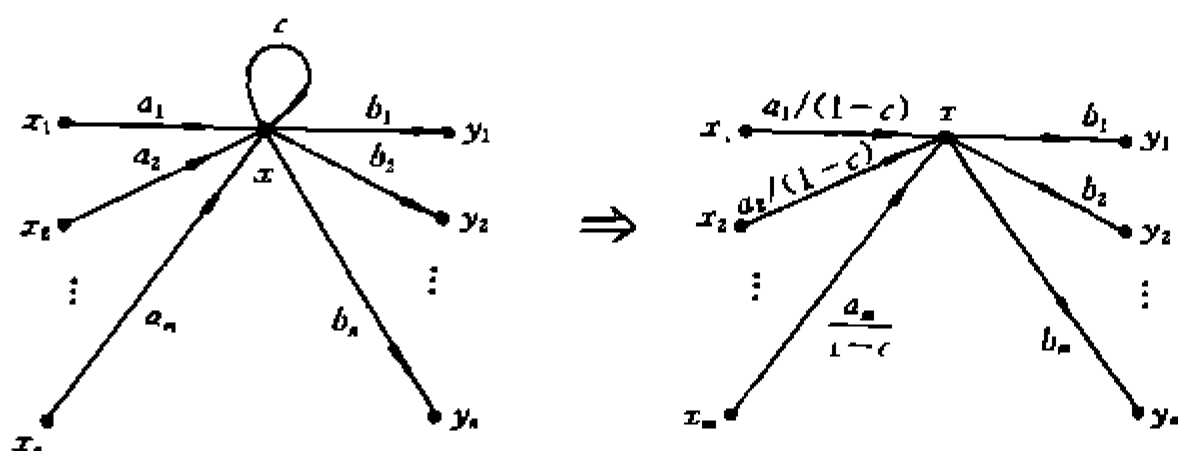


图 5-3-9

例1：

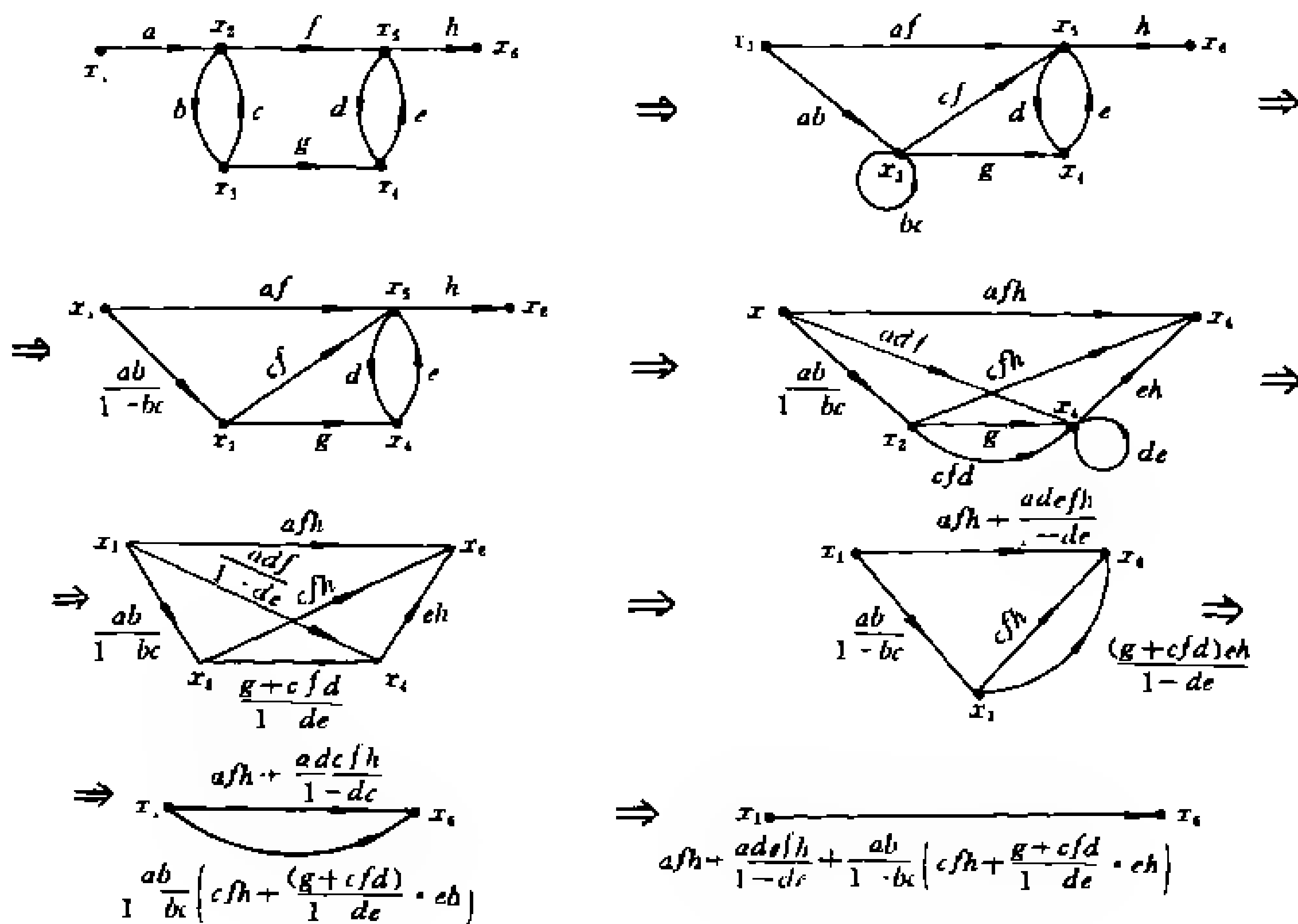


图 5-3-10

(5) 反向

以 $x_1=ax_2+bx_3$ 为例，$x_2=\frac{1}{a}x_1-\frac{b}{a}x_3$，

故有：

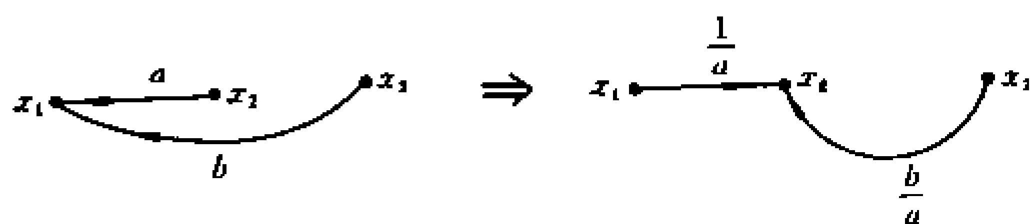


图 5-3-11

即对于源点 x_2，有向边 (x_2,x_1) 改为有向边 (x_1,x_2) 但权 a 改为 $\frac{1}{a}$，而有向边 (x_3,x_1) 改为 (x_3,x_2)，而其权改为 $\frac{b}{a}$。

同样理由可推广到一般：

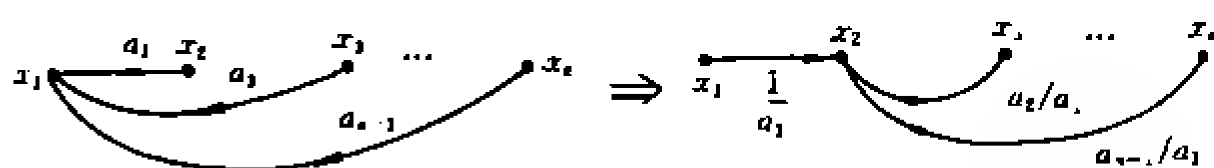


图 5-3-12

不难证明反向只能对于有向边 (x_i,x_j) 的始点 x_i 为源点时才有效。

例2：对方程组

$$\begin{cases} x_2=ax_1+bx_3, \\ x_4=cx_2+dx_3+ex_5, \\ x_6=fx_4+gx_5+hx_7, \end{cases}$$

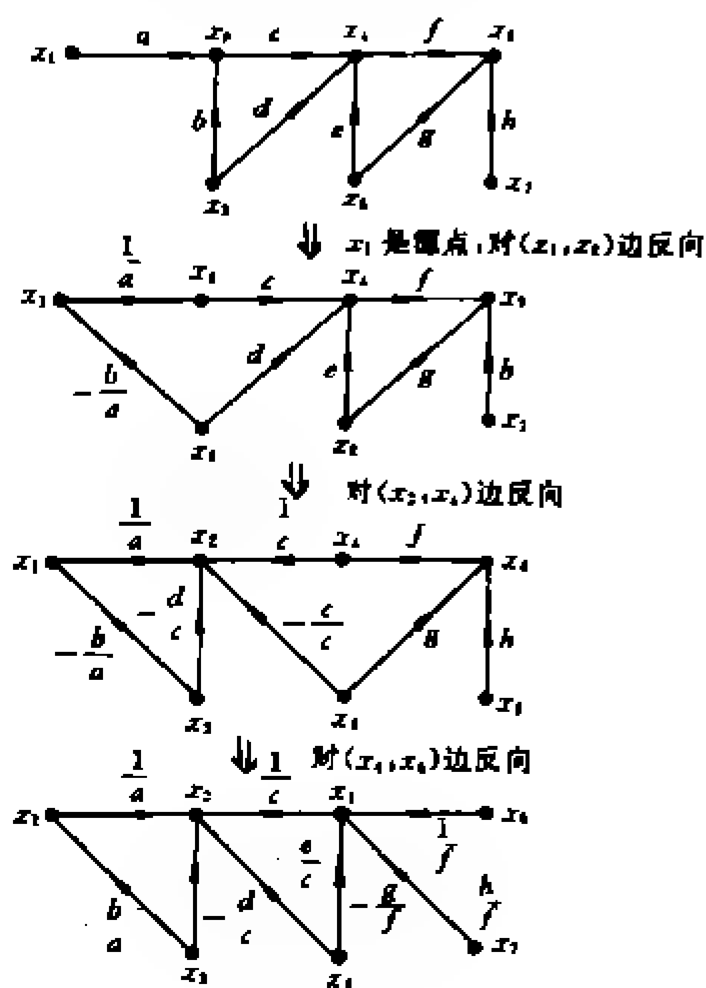


图 5-3-13

例3：

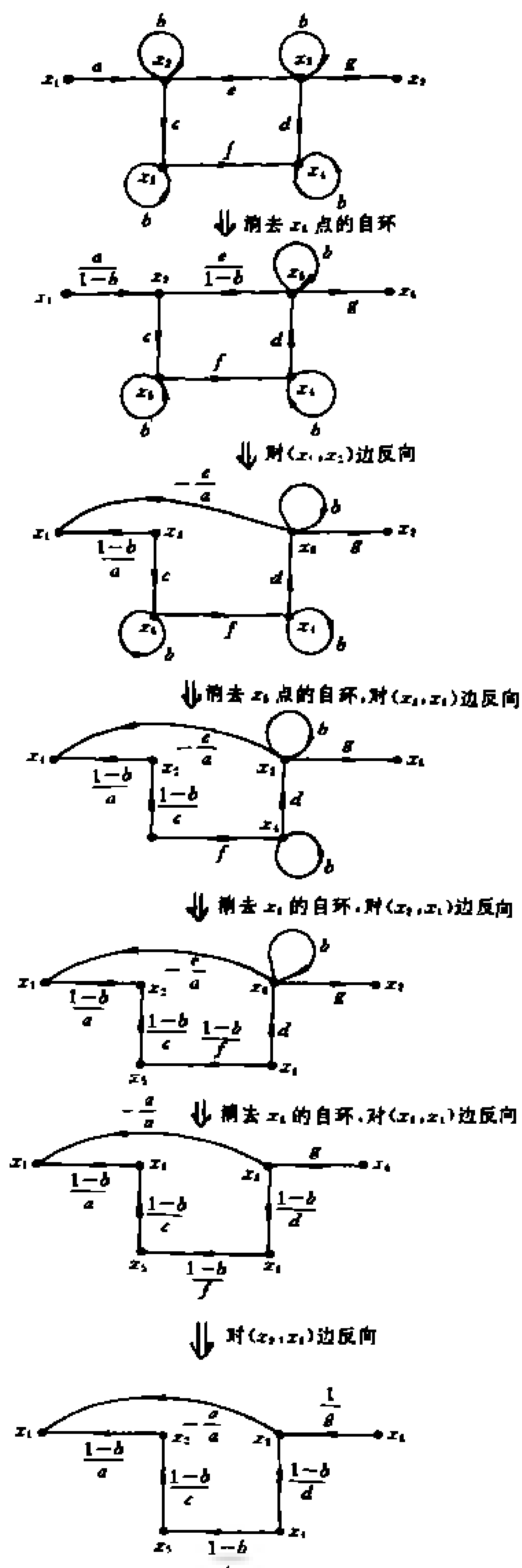

消去 x_2 点的自环
对 (x_1, x_2) 边反向
消去 x_5 点的自环，对 (x_2, x_5) 边反向
消去 x_4 的自环，对 (x_5, x_4) 边反向
消去 x_3 的自环，对 (x_4, x_3) 边反向
对 (x_3, x_6) 边反向

图 5-3 14

$$\therefore \qquad X_1 = \frac{1}{g}\left[\frac{(1-b)^4}{acdf} - \frac{e}{a}\right]x_6。$$

初学者务必注意图的演算过程的代数意义，这样才不至于发生似是而非的错误。

§4 行列式的展开法

行列式展开法：若 $A=(a_{ij})_{n\times n}$ 有

$$\det A = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & \cdots\cdots & \cdots\cdots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} = \sum_{(j)} \epsilon_{j_1 j_2 \cdots j_n} a_{1j_1} a_{2j_2} \cdots a_{nj_n}。$$

其中

$$\epsilon_{j_1 j_2 \cdots j_n} = \begin{cases} 1, & \text{若 } j_1 j_2 \cdots j_n \text{ 为偶置换,} \\ -1, & \text{若 } j_1 j_2 \cdots j_n \text{ 为奇置换。} \end{cases}$$

把次序 $j_1 j_2 \cdots j_n$ 改为自然顺序 $12\cdots n$ 所需要的置换次数叫做 $j_1 j_2 \cdots j_n$ 的置换次数。置换次数为奇数的叫做奇置换；否则叫做偶置换。置换奇偶数可以通过这个序列中大小倒置的个数来计算。例如23154，大小倒置有21、31、54三个。432561则有43、42、41、32、31、21、51、61。第一个例子是奇置换，后者为偶置换。12345也属偶置换。

下面给出一个定理，它是行列式展开的图算的依据。设 $A=(a_{ij})_{n\times n}$，G 是矩阵 A 的 Coates 图。

定理

$$\det A = (-1)^n \sum_{j=1}^{h} (-1)^{n_j} \triangle_j。$$

其中 n 为矩阵的阶，h 是图 G 中由过 n 个顶点的简单回路（不自身交叉的回路）形成的子图 G_j 的个数，n_j 是 G_j 中回路的数目，$\triangle_j$ 是 G_j 的各边的权的积。

在证明这定理之前，先就定理的意思通过例子解释清楚。

设

$$A = \begin{pmatrix} -1 & 2 & 1 \\ 1 & 2 & 1 \\ -1 & 1 & -1 \end{pmatrix}。$$

它的图（见图5-4-1）。

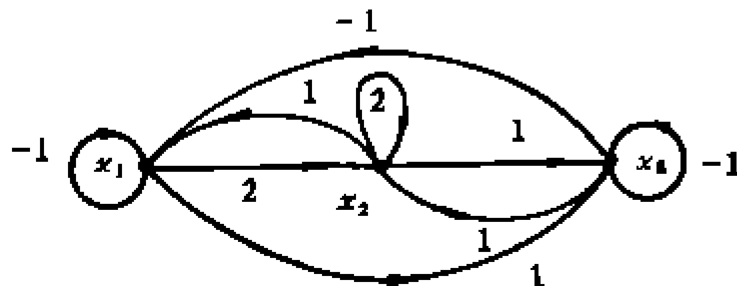


图 5-4-1

过 x_1, x_2, x_3 点的简单回路有以下六个，如图5-4-2所示的 $G_1, G_2, G_3, G_4, G_5, G_6$，故 $h=6$，$n=3$.

$j=1$ 时 $n_1=3$，

$\triangle_1=(-1)(-1)(2)=2$；

$j=2$ 时 $n_2=2$，

$\triangle_2=(-1)(+1)(2)=-2$；

$j=3$ 时 $n_3=2$，

$\triangle_3=(-1)(1)(2)=-2$；

$j=4$ 时 $n_4=2$，

$\triangle_4=(1)(1)(-1)=-1$；

$j=5$ 时 $n_5=1$，

$\triangle_5=(2)(1)(-1)=-2$；

$j=6$ 时 $n_6=1$，

$\triangle_6=(1)(1)(1)=1$。

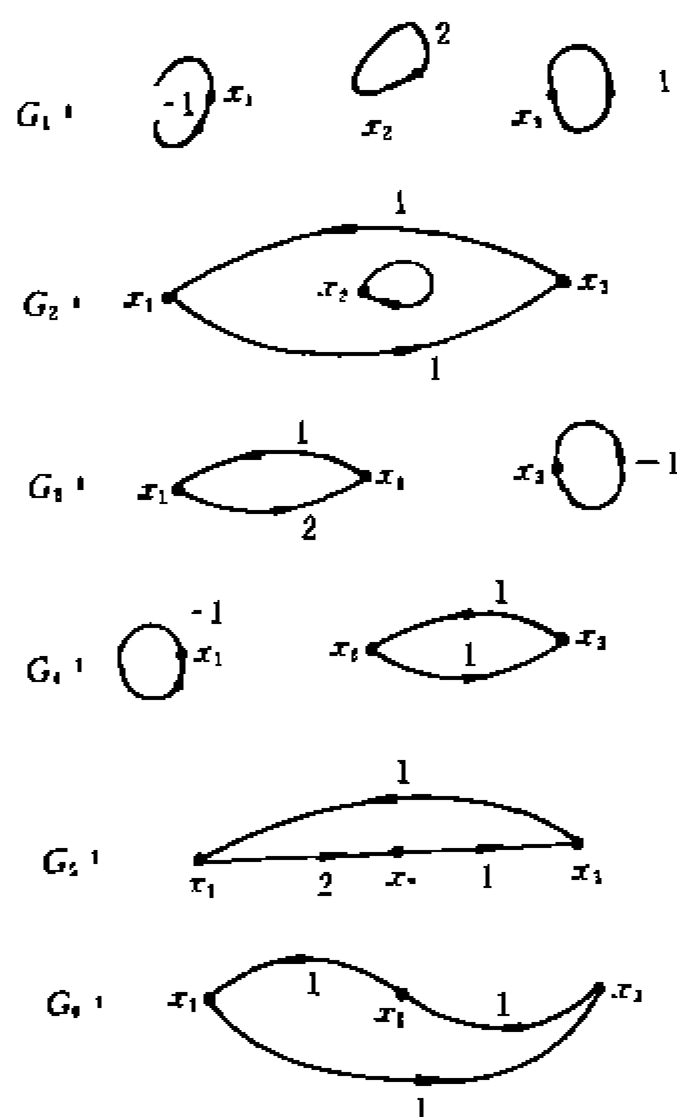


图 5-4-2

$$\therefore \quad D=(-1)^3\sum_{j=1}^{6}(-1)^{n_j}\triangle_j$$

$$=-(-2-2-2-1+2-1)$$

$$=6,$$

$$\begin{vmatrix} -1 & 1 & -1 \\ 2 & 2 & 1 \\ 1 & 1 & -1 \end{vmatrix}$$

$=2-2+1+2+1+2$

$=6$

定理的证明：

对于行列式展开式中每一项：

$$a_{1j_1}a_{2j_2}\cdots\cdots a_{nj_n}$$

对应了一个 G_j；反过来说每一个 G_j 都对应这样的一项，只要证明符号也是对的就可以了。

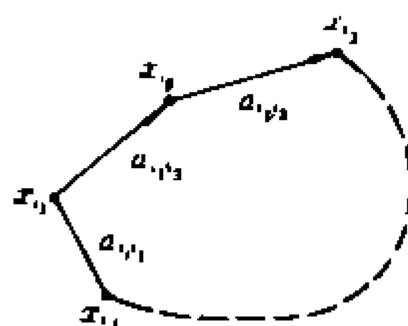


图 5-4-3

对于其中长度为 l 的回路：

要使得下标序列 $i_2i_3\cdots i_li_1$ 换成下标序列 $i_1i_2\cdots i_l$，只要 $l-1$ 次置换就可以办到。

如果子图 G_i 由 j 个回路组成，则顶点下标的排列次序为：

$$\underbrace{i_1, i_2, \cdots, i_{l_1}}_{\text{第一个回路}},\ \underbrace{j_1, j_2, \cdots, j_{l_2}}_{\text{第二个回路}},\cdots\cdots,\underbrace{k_1, k_2, \cdots, k_l}_{\text{第 } j \text{ 个回路}}$$

对应于这 G_i 的行列式中的项为：

$$\varepsilon a_{i_1i_2}a_{i_2i_3}\cdots a_{i_{l_1}i_1}a_{j_1j_2}a_{j_2j_3}\cdots\cdots a_{k_1k_2}a_{k_2k_3}\cdots a_{k_{l_j}k_1}$$

其中 $\varepsilon=\pm1$，$l_1+l_2+\cdots+l_t=n$。要把这式中的第二个下标序列 $i_2i_3\cdots i\ j_2j_3\cdots k_{l_t}k_1$ 改变为第一个下标序列 $i_1i_2\cdots j_1\cdots k_{l_t}$ 只要作

$$
\begin{aligned}
&(l_1-1)+(l_2-1)+\cdots+(l_t-1)\\
&=(l_1+l_2+\cdots+l_t)-t=n-t
\end{aligned}
$$

次置换。所以取 $\varepsilon=(-1)^{n-t}=(-1)^{n-t+2t}=(-1)^{n+t}$。

§5 代数方程组的 Coates 图解法

设方程组为：

$$
\begin{cases}
a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n=b_1,\\
a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n=b_2,\\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n=b_n。
\end{cases}
$$

根据 Cramer 法则可得解为：

$$
x_i=\frac{1}{D}\sum_{j=1}^{n}b_jA_{ji},\ i=1,2,\cdots,n。
$$

其中 A_{ji} 为元素 a_{ji} 的代数余子式。求解这方程组的过程相当于除了计算系数行列式外，尚得计算下列 n 个行列式：

$$
\begin{vmatrix}
b_1 & a_{12} & \cdots & a_{1n}\\
b_2 & a_{22} & \cdots & a_{2n}\\
\cdots & \cdots & \cdots & \cdots\\
b_n & a_{n2} & \cdots & a_{nn}
\end{vmatrix},\quad
\begin{vmatrix}
a_{11} & b_1 & a_{13} & \cdots & a_{1n}\\
a_{21} & b_2 & a_{23} & \cdots & a_{2n}\\
\cdots & \cdots & \cdots & \cdots & \cdots\\
a_{n1} & b_n & a_{n3} & \cdots & a_{nn}
\end{vmatrix}
\cdots\cdots
\begin{vmatrix}
a_{11} & a_{12} & \cdots & a_{1,n-1} & b_1\\
a_{21} & a_{22} & \cdots & a_{2,n-1} & b_2\\
\cdots & \cdots & \cdots & \cdots & \cdots\\
a_{n1} & a_{n2} & \cdots & a_{n,n-1} & b_n
\end{vmatrix}。
$$

这些行列式的计算相当于对矩阵：

$$
A_0=\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} & b_1\\
a_{21} & a_{22} & \cdots & a_{2n} & b_2\\
\cdots & \cdots & \cdots & \cdots & \cdots\\
a_{n1} & a_{n2} & \cdots & a_{nn} & b_n\\
0 & 0 & \cdots & 0 & 0
\end{bmatrix}
$$

作如下的运算：

(1) 令 $a_{n+1,i}=1$，

(2) 第 i 列与第 $n+1$ 列互换，

(3) 求所得的矩阵的行列式的值。

设 A_0 矩阵的 Coates 图为 G_0，由于 A_0 的第 $n+1$ 行元素全部为零，故从 $x_1,x_2,\cdots,x_n$ 诸点到 x_{n+1} 点的边不存在。但从 x_{n+1} 点向 $x_1,x_2,\cdots,x_n$ 诸点的边是存在的。作运算：$a_{n+1,i}=1$，相当于从 x_i 点到 x_{n+1} 点引一条权为1的边，然后按上节所讨论的方法计算行列式的值。

例1：设有方程组

$$
\begin{pmatrix}
a_{11} & 0 & a_{13} & 0 & a_{15} \\
0 & a_{22} & a_{23} & 0 & 0 \\
a_{3} & a_{32} & 0 & a_{34} & a_{35} \\
a_{41} & 0 & a_{43} & 0 & a_{45} \\
0 & 0 & a_{53} & 0 & a_{55}
\end{pmatrix}
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}
=
\begin{pmatrix} b_1 \\ 0 \\ 0 \\ 0 \\ b_5 \end{pmatrix}。
$$

矩阵 A 和加边矩阵 A_0分别是：

$$
A = \begin{pmatrix}
a_{11} & 0 & a_{13} & 0 & a_{15} \\
0 & a_{22} & a_{23} & 0 & 0 \\
a_{31} & a_{32} & 0 & a_{34} & a_{35} \\
a_{41} & 0 & a_{43} & 0 & a_{45} \\
0 & 0 & a_{53} & 0 & a_{55}
\end{pmatrix}，
$$

$$
A_0 = \begin{pmatrix}
a_{1} & 0 & a_{13} & 0 & a_{15} & b_1 \\
0 & a_{22} & a_{23} & 0 & 0 & 0 \\
a_{31} & a_{32} & 0 & a_{34} & a_{35} & 0 \\
a_{41} & 0 & a_{43} & 0 & a_{45} & 0 \\
0 & 0 & a_{53} & 0 & a_{55} & b_5 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}。
$$

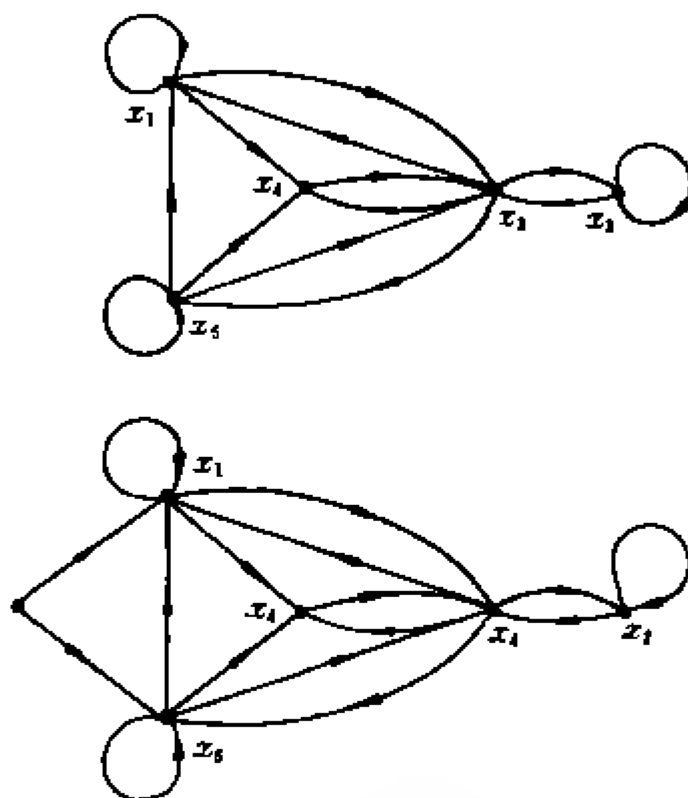


图 5-5-1

图5-5-1中从 x_j 到 x_i 的有向边的权为 a_{ij}，不 · ·说明。

用前面介绍的方法，当 $i=4$时可得：

$$
\begin{aligned}
x_4 = (&- b_1a_{41}a_{32}a_{23}a_{55} - b_1a_{31}a_{43}a_{22}a_{55} + b_1a_{31}a_{53}a_{45}a_{22} \\
&- b_1a_{41}a_{35}a_{53}a_{22} - b_5a_{45}a_{23}a_{32}a_{11} - b_5a_{45}a_{13}a_{31}a_{22} \\
&- b_5a_{35}a_{23}a_{11}a_{22} + b_5a_{35}a_{13}a_{41}a_{22} + b_5a_{15}a_{21}a_{32}a_{23}
\end{aligned}
$$

$$+ b_5a_{15}a_{31}a_{43}a_{22})/(a_{11}a_{22}a_{34}a_{43}a_{55} - a_{11}a_{22}a_{34}a_{45}a_{53} + a_{22}a_{43}a_{34}a_{53}a_{15} - a_{22}a_{41}a_{34}a_{13}a_{55})。$$

§6 Mason 公 式

假定我们所讨论的方程组具有如下形式：

$$\begin{cases} x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 + x_2 + \cdots + a_{2n}x_n = b_2, \\ \cdots\cdots\cdots\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + x_n = b_n。 \end{cases} \tag{1}$$

或写成：

$$\boldsymbol{AX} = \boldsymbol{B},$$

其中 $\boldsymbol{A}=(a_{ij})_{n\times n}$，但 $a_{jj}=1$　$j=1,2,\cdots,n$。

即

$$\boldsymbol{A} = \begin{pmatrix} 1 & a_{12} & \cdots & a_{1n} \\ a_{21} & 1 & \cdots & a_{2n} \\ \cdots & \cdots & & \\ a_{n1} & a_{n2} & \cdots & 1 \end{pmatrix}, \quad \boldsymbol{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \boldsymbol{B} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ x_n \end{pmatrix}。$$

可以把这方程组改写为：

$$\boldsymbol{X} = (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{X} + \boldsymbol{B} = \boldsymbol{A}_1\boldsymbol{X} + \boldsymbol{B},$$

$$\boldsymbol{A}_1 = \boldsymbol{I} - \boldsymbol{A}。$$

可以形象地把 $\boldsymbol{X}$ 看作下面系统中给以输入 B 后的输出。

B ⟹ [X = A X + B] ⟹ X

图　5.6.1

按§2叙述的方法作方程组(1)的 Mason 信号流图 G。

这个图 G 便称为是方程组 $AX=B$ 的 Mason 信号流图。下面介绍 Mason 公式如下。它对于许多问题，特别是关于线性反馈系统的分析，独具特色。

Mason 定理

设 G 是方程组 $\boldsymbol{AX}=\boldsymbol{B}$ 的信号流图，$L_1,L_2,\cdots,L_t$ 分别是图 G 的回路，w_i 是回路 L_i 的增益(即 L_i 诸边的权的乘积)，则

(1) $$D=\det A=1-\sum_i w_i + \sum_{\substack{i,j \\ L_i,L_j\text{互不接触}}} w_iw_j - \sum_{\substack{i,j,k \\ L_i,L_j,L_k\text{互不接触}}} w_iw_jw_k + \cdots\cdots$$

即 $D=\det A=1-$(所有回路增益的数的和)+(两两不相接触的回路增益的乘积的和)　……

(2) $x_i = \frac{1}{D}\sum_{j=1}^{n} b_j T_{ji}$，$i = 1,2,\cdots,n$。

其中　$T_{ji} = \sum_{(k)} W(P_k)\triangle_k$。

$W(P_k)$是从b_j点到x_i的道路P_k的增益；而$\triangle_k$为从D中消去与道路P_k上的点关联的所有项的结果。

下面举几个例子说明 Mason 公式。

例1：

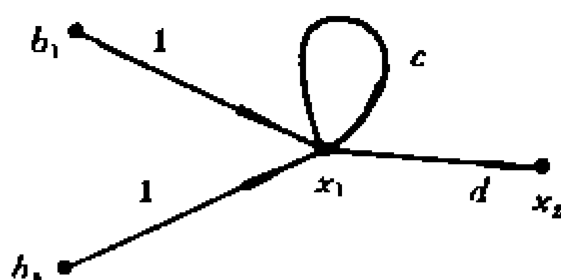


图　5-6-2

如图5-6-2。现用 Mason 公式解x_2于下。这里有两个自由项b_1和b_2。

由b_1到x_2的道路为$b_1x_1x_2$，其增益为d，同理由b_2到x_2的道路为$b_2x_1x_2$，其增益为d。这个图中仅有一个回路其增益为c，故

$$D = 1 - c$$

由于x_1点出现在D中，故$\triangle_1 = 1$，

这样便有：

$$x_2 = (b_1d + b_2d)/(1 - c)。$$

例2：如图5-6-3

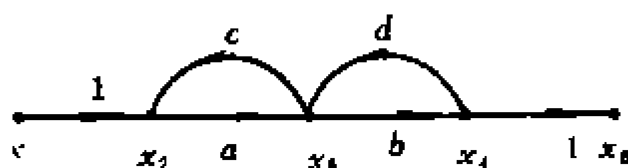


图　5-6-3

求x_5。

自由项为x_1，从x_1到x_5的道路为$x_1x_2x_3x_4x_5$，它的增益为ab。

这图中的回路有$x_2x_3x_2$，$x_3x_4x_3$，然而这两个回路在x_3点相接触，故

$$D = 1 - (ac + bd)，\triangle_1 = 1，$$

$$x_5 = \frac{x_1ab}{1 - ac - bd}。$$

例3：如图5-6-4，求x_5。

从x_1到x_5的道路有二：$x_1x_2x_3x_4x_5$，$x_1x_2x_5$。回路有二：$x_2x_3x_2$，$x_3x_4x_3$，$x_2x_3x_4x_2$，故

$$D = 1 - (bc + de + bdg)，$$

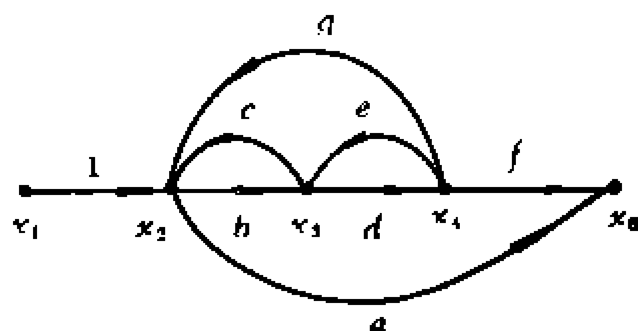


图 5-6 4

∴
$$x_5 = \frac{bdf + h(1 - de)}{1 - (bc + de + bdg)}x_1。$$

例4：如图5-6-5

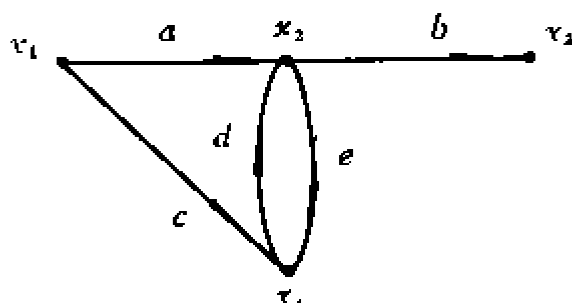


图 5 6 5

求 x_3。

从 x_1到 x_3的道路有二：$x_1x_2x_3$，$x_1x_4x_3$，回路有一：$x_2x_4x_2$，故

$$x_3 = \frac{ab + bcd}{1 - de}x_1。$$

例5：有方程组：

$$\begin{cases} x_1 - x_3 + b, \\ x_2 - 2x_1 + 3x_2, \\ x_3 = 2x_3 + x_4, \\ x_4 - x_1 + x_2。\end{cases}$$

求 x_4。

这方程组的图是：

从 b 到 x_4点的道路有二：bx_1x_4，$bx_1x_2x_4$，其增益分别为1和2，回路有下面四个：
它们的增益分别为3、2、1和2。

不相接触的回路有 c_1 与 c_2、c_1 与 c_3，不存在三个不相接触的回路。故

$D - 1 \quad (3 + 2 + 1 + 2) + (6 + 3) = 2$，

$\triangle_1 = 1 - (3 + 2) + 6 = 2$，

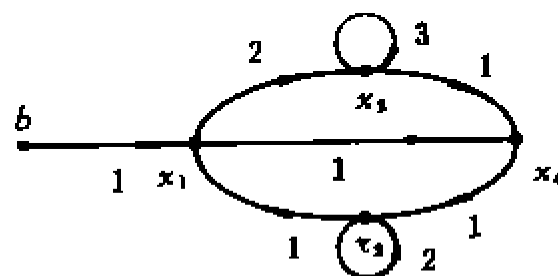


图 5-6-6

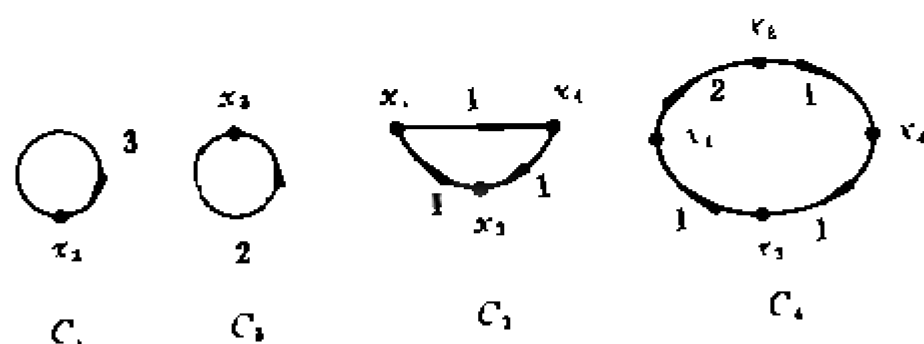


图 5-6-7

$$\triangle_2 = 1 - 2 = -1。$$

∴ $$x_4=\frac{b}{2}[2+2(-1)]=0。$$

例6：已知方程组

$$\begin{cases} x_1 - x_2 + x_3 = 0, \\ 2x_1 + 2x_2 + x_3 = 0, \\ x_1 + x_2 - x_3 = -b。 \end{cases}$$

求 x_1。

将这个方程组改写成如下形式：

$$\begin{cases} x_1 = x_2 - x_3, \\ x_2 = 2x_1 + 3x_2 + x_3, \\ x_3 = x_1 + x_2 + b。 \end{cases}$$

这个方程组的图如图5-6-8所示。

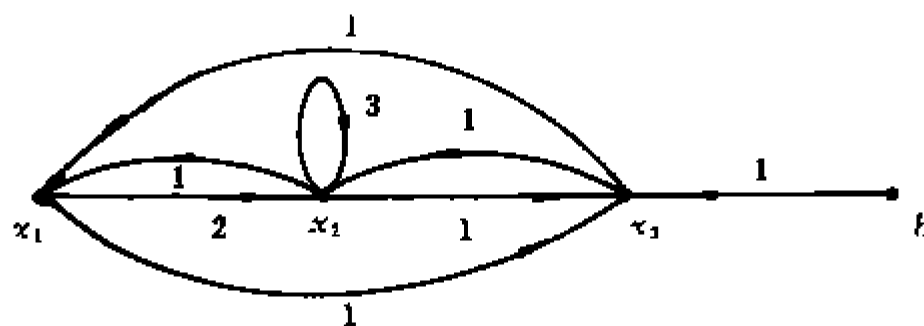


图 5-6-8

这图的回路有6种（见图5-6-9）。

两两不相接触的回路有 c_1、c_4。

所以：

$$D = 1 - [(-1) + 1 + 2 + 3 + 1 + (-2)] + (-1)(3)$$
$$= -6$$

从 b 到 x_1 的道路有以下两条：

故：$$x_1 = \frac{b}{-6}[(1)(1)+(-1)(-2)]$$
$$= -\frac{1}{2}b.$$

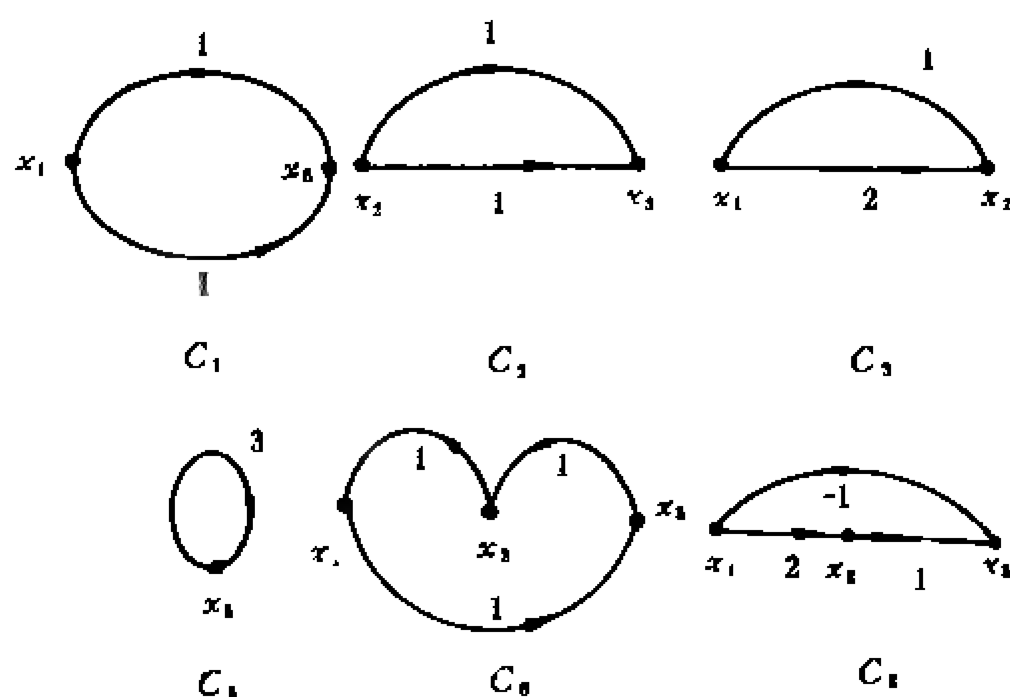


图 5.6.9

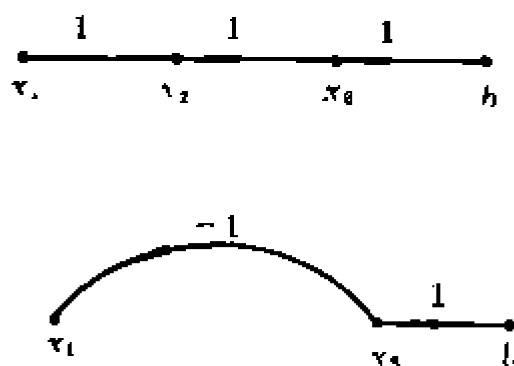


图 5.6.10

*§7 Mason 公式的证明

Mason 公式的证明方法颇多，但都十分繁琐，下面的证明方法是其中最好的一个。

不失一般性下约定方程组 $AX=B$ 的系数矩阵 A，其主对角元素均为1，即：$a_{ii}=1$，$i=1, 2, \cdots, n$。

$G=(V,E)$ 是方程组 $AX=B$ 的 Mason 信号流图。

定义 矩阵：$M=(m_{ij})_{n\times m}$，$D=(d_{ij})_{n\times m}$，$W=(w_{ij})_{m\times m}$。其定义如下：

$$m_{ij} \triangleq \begin{cases} 1, & (x_i, x_k)=e_j \in E; \\ 0, & \text{其它。} \end{cases}$$

$$d_{ij} = \begin{cases} 1, & (x_k, x_i)=e_j \in E; \\ 0, & \text{其它。} \end{cases}$$

$$w_{ij} = \begin{cases} 0, & i \neq j, \\ w_i, & i = j. \end{cases}$$

---

* 建议留作自学。

其中 w_i 是第 i 条边的权。

显然，若 $\boldsymbol{B}$ 是 G 图的邻接矩阵，则 $\boldsymbol{B}-\boldsymbol{M}-\boldsymbol{D}$。

例1：

$$\begin{cases} x_1 - f x_2 \quad e x_3 = b_1, \\ \quad a x_1 + x_2 - d x_4 - 0, \\ - b x_1 + x_3 \quad g x_4 = b_3, \\ - c x_3 + x_4 - 0。\end{cases}$$

下面是这方程组的图

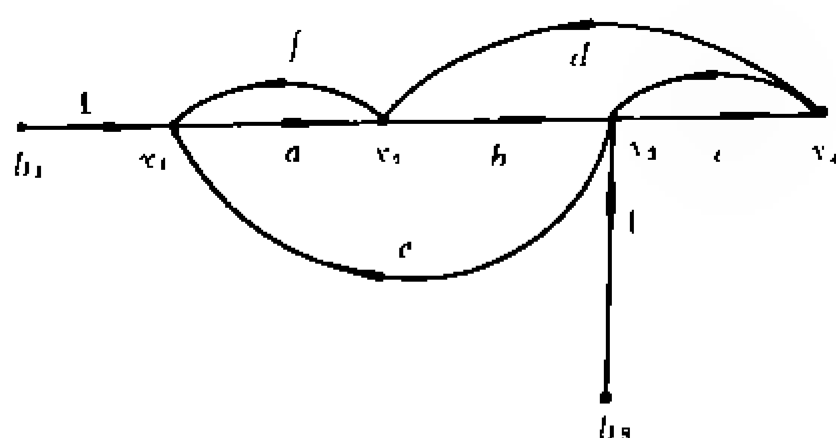


图 5-7 1

$$\boldsymbol{M}=\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} x \\ x_2 \\ x_3 \\ x_4, \end{matrix}$$
$$\quad a \quad b \quad c \quad d \quad e \quad f \quad g$$

$$\boldsymbol{D}-\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} x \\ x_2 \\ x_3 \\ x_4, \end{matrix}$$
$$\quad a \quad b \quad c \quad d \quad e \quad f \quad g$$

$$\boldsymbol{W}-\begin{pmatrix} a & & & & & & \\ & b & & & \boldsymbol{O} & & \\ & & c & & & & \\ & & & d & & & \\ & & & & e & & \\ & \boldsymbol{O} & & & & f & \\ & & & & & & g \end{pmatrix} \begin{matrix} a \\ b \\ c \\ d \\ e \\ f。 \\ g \end{matrix}$$
$$\quad a \quad b \quad c \quad d \quad e \quad f \quad g$$

定理 若 G 是 $\boldsymbol{AX}=\boldsymbol{B}$ 的 Mason 信号流图，则

$$\boldsymbol{I}_{(n)} - \boldsymbol{DWM}^T - \boldsymbol{A}。$$

证明 令 q_{ij} 是矩阵 $I_n - DWM^T$ 的第 i 行第 j 列元素，当 $i\neq j$ 时，显然有：

$$q_{ij} = -\sum_{k=1}^{m}\sum_{r=1}^{m} d_{ik}w_{kr}m_{jr}。$$

由于 W 是对角线矩阵，故

$$q_{ij} = -\sum_{k=1}^{m} d_{ik}w_{kk}m_{jk}。$$

由 M, W, D 矩阵的定义，

$$d_{ik}w_{kk}m_{jk} = w_{kk} = \begin{cases} w_k,\ (x_j,\ x_i) = e_k \\ 0,\ 其它 \end{cases}$$

所以
$$q_{ij} = a_{ij}.\quad (i\neq j)$$

当 $i=j$ 时，因没有自环，故

$$d_{ik}w_{kk}m_{ik} = 0,\ k = 1,\ 2,\ \cdots,\ m。$$

所以

$$q_{ii} = 1 - \sum_{k=1}^{m} d_{ik}w_{kk}m_{ik} = a_{ii} - 1。$$

引理1 由顶点 x_{i_1}、x_{i_2}、$\cdots$、x_{i_k} 和边 e_{j_1}，e_{j_2}，$\cdots$，e_{j_k} 所确定的 M(或 D)的 k 阶子阵 S 非奇异的充分必要条件是：(x_{i_1}，x_{i_2}，$\cdots$，x_{i_k})中每一顶点有一条且仅有一条(e_{j_1}，e_{j_2}，$\cdots$，e_{j_k})中的边从该点出发或进入，并且 $\det S=\pm 1$。

证明 若顶点集合(x_{i_1}，x_{i_2}，$\cdots$，x_{i_k})中每一个顶点都有边集合(e_{j_1}，e_{j_2}，$\cdots$，e_{j_k})中一条边从(或向)这点出发(或进入)，则每行都有一元素1。由于每一条边 e_{j_l} 只与一个顶点关联，故每列中只有一个非零元素1。所以 S 矩阵的每行每列都有一个且仅有一个非零元素 ±1。故 $\det S=\pm1$。即 S 非奇异。

反之，若 S 非奇异，则每行至少必有一非零元素1，且仅有一个非零元素1。如若不然，由于每一条边只与一个顶点关联，故最多只有 k 个1，如若有一行有一个以上的非零元素 ±1，则必有一行全为0。这与 S 非奇异的假设发生矛盾。引理1证毕。

定理 由顶点 x_{i_1}，x_{i_2}，$\cdots$，x_{i_k} 和边 e_{j_1}，e_{j_2}，$\cdots$，e_{j_k} 所确定的分别为 M 和 D 的子方阵 S_1 和 S_2，都非奇异的充分必要条件是边 e_{j_1}，e_{j_2}，$\cdots$，e_{j_k} 形成一回路，或一组不相接触的回路。

证明 由引理1知 S_1、S_2 非奇异的充分必要条件是(x_{i_1}，x_{i_2}，$\cdots$，x_{i_k})中每一点都有(e_{j_1}，e_{j_2}，$\cdots$，e_{j_k})中的边进入和发出。无论如何，一个子图的每一点都有一边进来，一边出去，且仅有一条边进去，也仅有一条边出去时，只能是回路，或一组不相接触的回路。定理证毕。

例如从本节的例1来看由 b、c、d 边形成的回路，对应于 M 和 D 的子阵：

$$S_1 = \begin{matrix} x_2 \\ x_3 \\ x_4 \end{matrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{\ b\ \ c\ \ d},\qquad S_2 = \begin{matrix} x_2 \\ x_3 \\ x_4 \end{matrix}\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}_{\ b\ \ c\ \ d}。$$

显然 S_1、S_2 都是非奇异的。

引理2 设 S_1 和 S_2 分别是 M、D 中相对应的非奇异子阵。设 L_1、L_2、$\cdots$、L_r 为由 S_1 和 S_2 的列对应的边组成的互不接触的回路。p 为集合(L_1，L_2，$\cdots$，L_r)中有偶数个边的回路的数

目。则S_1和S_2的行列式相等的充分必要条件是p为偶数。

在证明引理2以前，先通过一个例子来加以说明。在本节开始的例1中，下面两个矩阵S_1和S_2分别是M和D中相对应的子阵：

$$S_1=\begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \ \end{matrix}\begin{matrix}\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}\\ \begin{matrix}a&c&f&g\end{matrix}\end{matrix},\qquad S_2=\begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \ \end{matrix}\begin{matrix}\begin{pmatrix}0&0&1&0\\1&0&0&0\\0&0&0&1\\0&1&0&0\end{pmatrix}\\ \begin{matrix}a&c&f&g\end{matrix}\end{matrix}。$$

显然(a,c)，(f,g)为互不接触的两回路。不难证明：

$$\det S_1=\det S_2=-1。$$

证明 设回路L_i的边和顶点的编排次序使得边$e_{i_1},e_{i_2},\cdots,e_{i_{n_i}}$和顶点$v_{i_1},v_{i_2},\cdots,v_{i_{n_i}}$有这样的关系：边$l_{i_j}$从$v_{i_j}$出发而进入$v_{i_{j+1}}$。$S_1$中由这些边和顶点对应的子阵$S_{1i}$，$S_{1i}$在$S_2$中的对应子阵为$S_{2i}$分别如下：

$$S_{1i}=\begin{matrix} v_{i_1} \\ v_{i_2} \\ \vdots \\ v_{i_{n_i}} \\ \ \end{matrix}\begin{matrix}\begin{pmatrix}1&0&\cdots&0\\0&1&\cdots&0\\\cdots&\cdots&\cdots&\cdots\\0&0&\cdots&1\end{pmatrix}\\ \begin{matrix}e_{i_1}&e_{i_2}&&e_{i_{n_i}}\end{matrix}\end{matrix},\qquad S_{2i}=\begin{matrix} v_{i_1} \\ v_{i_2} \\ \vdots \\ v_{i_{n_i}} \\ \ \end{matrix}\begin{matrix}\begin{pmatrix}0&0&\cdots&1\\1&0&\cdots&0\\\cdots&\cdots&\cdots&\cdots\\0&0&\cdots&0\end{pmatrix}\\ \begin{matrix}e_{i_1}&e_{i_2}&&e_{i_{n_i}}\end{matrix}\end{matrix}。$$

显然S_{1i}和S_{2i}都是非奇异矩阵，而S_{2i}必须作n_i-1次换行变换才能和S_{1i}一致。故

$$\det S_{1i}=\det S_{2i}，$$

当而且仅当n_i是奇数。而且

$$\det S_{1i}=-\det S_{2i}$$

当而且仅当n_i为偶数。

若$L_1,L_2,\cdots,L_r$的排列次序是这样的，使得$L_1,L_2,\cdots,L_p$正好都是偶数边的回路，其中$p\leqslant r$，其余的都是奇数个边的回路。则

$$S_1=\begin{pmatrix}S_{11}&&&&&&\\&S_{12}&&&&\boldsymbol{O}&\\&&\ddots&&&&\\&&&S_{1p}&&&\\&&&&S_{1,p+1}&&\\&\boldsymbol{O}&&&&\ddots&\\&&&&&&S_{1r}\end{pmatrix},$$

$$S_2=\begin{pmatrix}S_{21}&&&&&&\\&S_{22}&&&&\boldsymbol{O}&\\&&\ddots&&&&\\&&&S_{2p}&&&\\&&&&S_{2,p+1}&&\\&\boldsymbol{O}&&&&\ddots&\\&&&&&&S_{2r}\end{pmatrix}。$$

$\therefore\ \det S_1=\det S_{11}\det S_{12}\cdots\det S_{1p}\det S_{1,p+1}\cdots\det S_{1r}，$

$\det S_2 = \det S_{21} \det S_{22} \cdots \det S_{2p} \det S_{2,p+1} \cdots \det S_{2r}$。

但

$$\det S_{1i} = -\det S_{2i},\quad i \leqslant p,$$
$$\det S_{1i} = \det S_{2i},\quad i > p。$$

所以

$$\det S_1 = (-1)^p \det S_2。$$

引理3 S_1和S_2分别是M和D矩阵中对应的非奇异子阵。$L_1, L_2, \cdots, L_r$是由它的列对应的边$e_1, e_2, \cdots, e_k$构成的不相接触的回路。则$\det S_1 = \det S_2$的充要条件是k和r同是奇数或同是偶数。

证明 设p是$(L_1, L_2, \cdots, L_r)$中边数为偶数的回路数目。若p为偶数，r也是偶数，则边数为奇数的回路数目$r-p$也必为偶数。所以边数k也必是偶数，现在把各种情况列表如下：

	p	$r-p$	r	k
1	偶	偶	偶	偶
2	偶	奇	奇	奇
3	奇	偶	奇	偶
4	奇	奇	偶	奇

从上表可知，仅当p为偶数时，r与k同为偶数，或同为奇数。根据引理2，当且仅当n为偶数时，S和S_2的行列式的值相同。

现在证Mason定理的第一部分。即

$$\det A = 1 - \sum_{i=1}^{r} w_i + \sum_{\substack{i,j \\ L_i, L_j \text{不接触}}} w_i w_j - \sum_{\substack{i,j,k \\ L_i, L_j, L_k \text{不接触}}} w_i w_j w_k + \cdots$$

证明用到一个行列式展开式：

$$\det(\lambda I + A) = \begin{vmatrix} a_{11} + \lambda & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} + \lambda & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} + \lambda & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} + \lambda \end{vmatrix}$$
$$= \lambda^n + s_1 \lambda^{n-1} + s_2 \lambda^{n-2} + \cdots + s_{n-1} \lambda + s_n。$$

其中：

$$s_k = \sum_{q_1 < q_2 < \cdots < q_k} \begin{vmatrix} a_{q_1 q_1} & a_{q_1 q_2} & \cdots & a_{q_1 q_k} \\ a_{q_2 q_1} & a_{q_2 q_2} & \cdots & a_{q_2 q_k} \\ \cdots & \cdots & \cdots & \cdots \\ a_{q_k q_1} & a_{q_k q_2} & \cdots & a_{q_k q_k} \end{vmatrix}$$

根据

$$\begin{vmatrix} a & b+c \\ a_2 & b_2+c_2 \end{vmatrix} = \begin{vmatrix} a & b \\ a_2 & b_2 \end{vmatrix} + \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}$$

展开

$$\begin{vmatrix} a_{11}+\lambda & a_{12}+0 & a_{13}+0 & \cdots & a_{1n}+0 \\ a_{21}+0 & a_{22}+\lambda & a_{23}+0 & \cdots & a_{2n}+0 \\ a_{31}+0 & a_{32}+0 & a_{33}+\lambda & \cdots & a_{3n}+0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1}+0 & a_{n2}+0 & a_{n3}+0 & \cdots & a_{nn}+\lambda \end{vmatrix}$$

即可得上面的等式。

令 $\boldsymbol{R}=\boldsymbol{DWM}^T$，则 $\boldsymbol{A}=\boldsymbol{I}-\boldsymbol{R}$。

所以

$$\det A = \det(I - R) = 1 + \sum_{j=1}^{n}(-1)^j s_j。$$

由于矩阵 $\boldsymbol{A}$ 的主对角线上元素为1，所以 $\boldsymbol{R}$ 的主对角线上元素为零。

不失一般性，考虑一典型的 k 阶子阵 $\boldsymbol{R}_k$，它由矩阵 $\boldsymbol{R}$ 的第1，2，…，k 行和第1，2，…，k 列元素组成的。则：

$$\det \boldsymbol{R}_k = \det(\boldsymbol{D}_0\boldsymbol{W}\boldsymbol{M}_0^T)。$$

其中 $\boldsymbol{D}_0$是从 D 中去掉第 $k+1$，…，n 行后的子阵；$\boldsymbol{M}_0$是从 $\boldsymbol{M}$ 中去掉相同的行后的结果。对

$$\det \boldsymbol{R}_k = \det((\boldsymbol{D}_0\boldsymbol{W})\boldsymbol{M}_0^T)$$

利用 Binet Cauchy 定理展开。但矩阵 $\boldsymbol{D}_0\boldsymbol{W}$ 是对 $\boldsymbol{D}_0$的每列乘以该列对应的边的权。故 $\boldsymbol{D}_0\boldsymbol{W}$ 的子方阵非奇异的充要条件是 $\boldsymbol{D}_0$的相应子方阵非奇异。故 $\det(\boldsymbol{D}_0\boldsymbol{W}\boldsymbol{M}_0^T)$展开式中的项为

$$\pm w_1 w_2 \cdots\cdots w_k。$$

其中 w_i 是 e_i 边的权；而且 e_1，e_2，…，e_k 形成一回路或不相接触的一系列的回路 L_1，L_2，…，L_r，这些回路和顶点 v_1，v_2，…，v_k 发生关联。除了符号以外，Mason 公式已得证明。现在把符号列表于下：

	k	r	$w_1w_2\cdots w_k$ 在 R_k 中符号（引理3）	R_k 在 A 中的符号$(-1)^k$	$w_1\cdots w_k$ 在 $\lvert A\rvert$ 的符号
1	偶	偶	+	+	+
2	偶	奇		+	—
3	奇	偶			+
4	奇	奇	+		

从上表可知，当且仅当回路数为偶数时取正号，当且仅当回路数为奇数时取负号。定理的第一部分证毕。

现在进而证明定理的第二部分。

不失一般性，假定我们讨论 A_{n1}，$n\neq1$，则

$$A_{n1}=(-1)^{n+1}\begin{vmatrix} a_{12} & a_{13} & \cdots & a_{1n} \\ & & & a_{2n} \\ & Q & & \vdots \\ & & & a_{n-1,n} \end{vmatrix},$$

其中

$$Q=\begin{vmatrix} 1 & a_{23} & \cdots & a_{2,n-1} \\ a_{32} & 1 & \cdots & a_{3,n-1} \\ \vdots & & & \vdots \\ a_{n-1,2} & a_{n-1,3} & \cdots & 1 \end{vmatrix},$$

所以

$$A_{n1}=(-1)^{n+1}\left[(-1)^{n}a_{1n}Q+(-1)^{n-1}\sum_{i,j=2}^{n-1}a_{in}a_{1j}Q_{ij}\right]。$$

其中 Q_{ij} 是 Q 中第 i 行第 j 列元素的代数余子式，

$$i,j=1,2,3,\cdots,n-1$$

令 $t_{ji}=-a_{ij}$，则 t_{ji} 是边 e_{ji} 的权。e_{ji} 表示从 x_j 点出发进入 x_i 点的边。作 $t_{ji}=a_{ij}$ 的置换得：

$$A_{n1}=t_{n1}Q+\sum_{i,j=2}^{n-1}t_{ni}t_{j1}Q_{ij}。$$

对应于 Q 的图 G'，是从图 G 中消去点 x_1 和 x_n 点，以及和这些点关联的边所得的子图，特别设有边 e_{ni} 和 e_{j1}。

$$Q_{ij}=t_{ij}V+\sum_{\substack{k,m=2\\k,m\neq i,j}}^{n-2}t_{ik}t_{mj}P_{km}。$$

其中对应于 V 的图 G'' 是从 G' 中去掉顶点 x_i 和 x_j 点，及与这两点关联的所有的边以后的子图。

继续上面的步骤可得 A_n 的典型项：

$$t_{ni}t_{ik}t_{kp}\cdots t_{qm}t_{mj}t_{j1}|R|。$$

其中 $t_{ni}\cdots t_{j1}$ 是从 x_n 到 x_1 点的道路 P_k 的权的乘积；而 $|R|$ 是从 G 中去掉 P_k 中所有的点，以及与这些点相关联的边以后的子图所表达的方程组的行列式。定理证毕。

对于一般的方程组，其中 a_{ii} 不一定为1时 Mason 公式仍然是对的，证明留作思考。

习 题

1. 画出下列方程组的信号流图，并利用 Mason 公式求 x_3 的解。

$$\begin{cases} ax_1-bx_2+cx_3=l, \\ dx_2-fx_3=0, \\ gx_1+hx_2+kx_3=m。 \end{cases}$$

2. 画出下列方程组的信号流图，并利用 Mason 公式求 x_3 的解。

$$\begin{cases} x_1 = ax_2 + dx_3 + 1, \\ x_2 = bx_1 + fx_2, \\ x_3 = cx_1 + ex_2 + gx_4, \\ x_4 = hx_3 + jx_5 + 1, \\ x_5 = ix_4. \end{cases}$$

3. 画出下列方程组的信号流图，并利用 Mason 公式求解：

$$\begin{pmatrix} 4 & 1 & 0 & 3 \\ 2 & 5 & 1 & 1 \\ 1 & 0 & 3 & -1 \\ 0 & 1 & 5 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}。$$

4. 解下列方程组

$$\begin{pmatrix} 3 & -2 & 1 \\ -1 & 2 & 0 \\ 3 & 2 & 2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix} y_4$$

5. 用§3中讨论的归约方法求上面问题1，3的解。

6. 用§3中讨论的归约方法求问题4的解。

7. 试用 Mason 流图法解方程组

$$\begin{pmatrix} 2 & 1 & 1 \\ 1 & -1 & 2 \\ 2 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} x_4。$$

第六章　网络流图问题

这一章我们以运输网络和开关网络为例，介绍网络的流及其它有关问题。网络流的理论在广泛的领域里具有重要的应用。

§1　网络流图问题与最大流

若有向图 $G=(V,E)$ 满足下列条件者，则称其为网络流图：

1. 有且仅有一个顶点 Z，它的入度为零，即 $d^-(Z)=0$，这个顶点 Z 便称为源，或称为发点。

2. 有且仅有一个顶点 $\bar{Z}$，它的出度为零，即 $d^+(\bar{Z})=0$，这个顶点 $\bar{Z}$ 便称为沟，或叫做收点。

3. 每一条边都有非负数，叫做该边的容量。边 (v_i, v_j) 的容量用 c_{ij} 表示。例如图6.1便是一个网络流图。

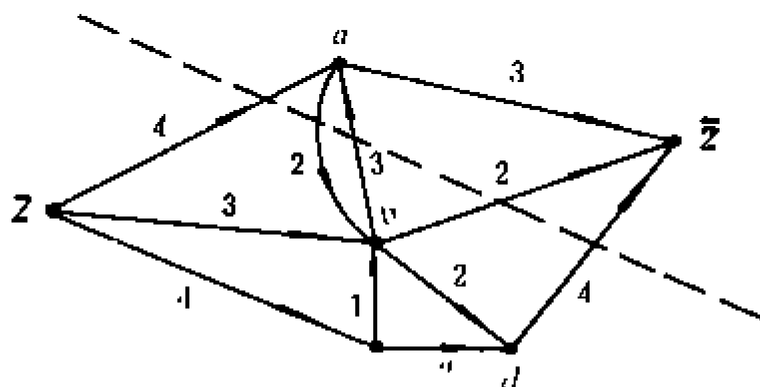


图 6.1

上面我们定义的网络常常称为运输网络。可用运输网络来表示的物理模型是多种多样的。网络的有向边可以用来表示城市之间的公路、电讯局之间的通讯线路；边的容量则可以是允许通过的物资、速率、公路上的汽车数量或信号流量等等。

对于网络流图 G，每一条边 (i,j) 都给定一个非负数 f_{ij}，这一组数满足下列两条件时称为这网络的容许流，用 f 表示它。

① 每一条边 (i,j) 有 $f_{ij} \leqslant c_{ij}$。　　　　(1)

② 除发点 Z 和收点 $\bar{Z}$ 以外的所有的点 v_i 恒有：

$$\sum_j f_{ij} = \sum_k f_{ki}$$

这个等式说明中间点 v_i 的流量守恒，输入与输出的量相等。

③ 对于源 Z 和沟 $\bar{Z}$ 有：

$$\sum_i f_{Zi} = \sum_j f_{j\bar{Z}} = w\text{。}$$

这个数 w 叫做这网络流的流量，而且从 Z 点发出的量等于进入收点 $\bar{Z}$ 的量。

当 $f_{ij}=c_{ij}$ 时便称边 (v_i, v_j) 为饱和，或 f 饱和，表示流 f 对该边已饱和。所谓最大流问题是求使得从发点 Z 送往收点 $\bar{Z}$ 的流量 w 达到最大的流 f。

运输网络的一个主要问题是要找出它的一个最大流 f。条件(1)是一个不等式，所以这个问题是一个典型的线性规划问题。现在我们以图 6-1-1 为例，说明为了使得从 Z 点送往 $\bar{Z}$ 点的流量 w 达到最大的数学问题的提法。关于 Z、a，b、c、d，$\bar{Z}$ 点的流量守恒关系，分别得：

$$\begin{cases} f_{Za}+f_{Zb}+f_{Zc}=w, \\ f_{Za}+f_{ba}=f_{a\bar{Z}}+f_{ab}, \\ f_{Zb}+f_{ab}+f_{cb}=f_{b\bar{Z}}+f_{bd}+f_{ba}, \\ f_{Zc}=f_{cb}+f_{cd}, \\ f_{bd}+f_{cd}=f_{d\bar{Z}}, \\ f_{a\bar{Z}}+f_{b\bar{Z}}+f_{d\bar{Z}}=w。 \end{cases}$$

$$\begin{cases} 0\leqslant f_{Za}\leqslant 4, \quad 0\leqslant f_{Zb}\leqslant 3, \quad 0\leqslant f_{Zc}\leqslant 4, \\ 0\leqslant f_{ab}\leqslant 2, \quad 0\leqslant f_{a\bar{Z}}\leqslant 3, \quad 0\leqslant f_{ba}\leqslant 3, \\ 0\leqslant f_{bd}\leqslant 2, \quad 0\leqslant f_{cb}\leqslant 1, \quad 0\leqslant f_{cd}\leqslant 2, \\ 0\leqslant f_{b\bar{Z}}\leqslant 2, \quad 0\leqslant f_{d\bar{Z}}\leqslant 4, \quad 0\leqslant f_{a\bar{Z}}\leqslant 3, w\geqslant 0。 \end{cases}$$

在满足上面条件下求 w 的最大值，即求

$$\max w。$$

如果把 w 看作是变量，则上面的问题是一个典型的线性规划问题；不过对这个具体问题来说，用图论的方法比较更为简洁有效。

§2 割　　切

为了解决求运输网络的最大流问题，这一节先来讨论割切的概念。

$G=(V,E)$ 是已知的网络流图，假定 S 是 V 的一个子集，而且 S 满足下面两个条件：

(a) $Z\in S$，　　　(b) $\bar{Z}\notin S$。

令 $\bar{S}=V\setminus S$，即 $\bar{S}$ 为 S 的补集。这样把顶点 V 便分成 S 和 $\bar{S}$ 两个部份，其中 $Z\in S$，$\bar{Z}\in \bar{S}$，对于一个端点在 S 而另一个端点在 $\bar{S}$ 的所有的边的集合叫做是割切，用 $(S,\bar{S})$ 表示。在这集合 $(S,\bar{S})$ 中，把从 S 到 $\bar{S}$ 的边的容量的和叫做这割切的容量，用 $C(S,\bar{S})$ 表示，即

$$C(S,\bar{S})=\sum_{\substack{i\in S\\ j\in \bar{S}}} c_{ij}$$

从第一节的例子可见，图 6-1-1 中虚线表示一个割切，其中：

$$S=\{Z,b,c,d\},\ \bar{S}=\{a,\bar{Z}\},$$

$$C(S,\bar{S})=c_{Za}+c_{ba}+c_{b\bar{Z}}+c_{d\bar{Z}}=4+3+2+4=13。$$

显然不同的割切有不同的割切容量。

定理　对于已知的网络流，从发点 Z 到收点 $\bar{Z}$ 的流量 w 的最大值小于或等于任何一个割切的容量，即

$$\max w \leqslant \min C(S,\bar{S})$$

证明 当 i 点既不是发点 Z，也不是收点 $\bar{Z}$ 的任意一点时，恒有：

$$\sum_{j\in V} f_{ij} - \sum_{j\in V} f_{ji} = 0。 \tag{1}$$

但 $i=Z$ 时有：

$$\sum_{j\in V} f_{Zj} = w \tag{2}$$

从(1)、(2)对 $i\in S$ 求和得

$$\sum_{i\in S,\ j\in V} [f_{ij} - f_{ji}] = w。$$

因为：

$$V = S\cup\bar{S},$$

所以

$$\sum_{i\in S,j\in V}[f_{ij}-f_{ji}] = \sum_{i\in S,j\in S}[f_{ij}-f_{ji}] + \sum_{i\in S,j\in \bar{S}}[f_{ij}-f_{ji}] = w$$

然而

$$\sum_{i\in S,j\in S} f_{ij} = \sum_{i\in S,j\in S} f_{ji}。$$

这等式成立是因为两个下标求和都是对 S 的全体进行的。即

$$\sum_{i\in S,j\in S} [f_{ij} - f_{ji}] = 0,$$

故有：

$$\sum_{i\in S,j\in \bar{S}} [f_{ij} - f_{ji}] = w。$$

但

$$0 \leqslant f_{ij} \leqslant c_{ij}$$

所以

$$f_{ij} - f_{ji} \leqslant f_{ij} \leqslant c_{ij}.$$

故

$$w = \sum_{i\in S,j\in \bar{S}} [f_{ij} - f_{ji}] \leqslant \sum_{i\in S,j\in \bar{S}} c_{ij} = C(S,\bar{S})。$$

因为割切 $(S,\bar{S})$ 是任意的，即流量 w 不大于任意一割切的流量。所以流量 w 小于或等于割切的流量的最小值。

故有：

$$\max_f w \leqslant \min_f C(S,\bar{S})。$$

§3 Ford-Fulkerson 最大流最小割切定理

前面曾经指出，运输网络中的一个主要问题就是寻求一网络流 f 使得流量 w 取得极大值。这个网络流 f 便称为最大流。上节证明了流量 w 不能超过割切容量的最小值，揭示了当流量等于某一割切容量时，这个流也必定是最大流。下面的 Ford Fulkerson 定理回答了这样的问题，即只有和某一割切容量相等时才是最大流量。也就是说和某割切容量相等是最大流的充要条件。自然这里所说的网络流都满足 $f_{ij}\leqslant c_{ij}$ 的条件，或说 f 是容许流。这个定理是网络流理论的核心。关于图的许多结果，在适当选择网络之后，应用这个定理往往能够容易地获得解决。Ford 和 Fulkerson 给出的证明是构造性的，可以从它引出求网络最大流的一个算法。

定理 在一个给定的网络流图上，流的极大值等于割切容量的最小值。

由于已经证明了：

$$\max w \leqslant \min C(S,\bar{S})$$

所以只需证明对于某一割切$(S,\bar{S})$，仅仅是$\max w<\min C(S,\bar{S})$是不够的。证明过程实际上是一种进一步提高流量的算法，直至使$\max w=\min C(S,\bar{S})$成立。为此定义网络流图G中，从Z到$\bar{Z}$的道路为网络流道路，即设

$$Z = v_0, v_1, \cdots, v_n = \bar{Z}$$

为网络流图G上点的序列。对于$i=0,1,2,\cdots,n-1$恒有(v_i,v_{i+1})或(v_{i+1},v_i)边是G的一条边时，则称$Z=v_0,v_1,\cdots,v_n=\bar{Z}$是一条从$Z$点到$\bar{Z}$点的道路。

图 6-3-1

由于图G是有向图，道路上的边的方向与道路方向一致与否分为两类，如图6 3 1所示若(v_i,v_{i+1})是G的边，则称它为向前边，即边的方向与道路方向一致；反之，若(v_{j+1},v_j)是G的边，则称之为后退边，后退边的方向与$Z\to\bar{Z}$的道路方向相反，也叫反向边。

对于向前边(i,j)，若$f_{ij}=c_{ij}$，或后退边(i,j)的流量$f_{ij}=0$，称之为饱和边。

对于从Z点到$\bar{Z}$点的道路上所有的向前边(i,j)恒有$f_{ij}<c_{ij}$；对于所有后退边(i,j)恒有$f_{ji}>0$，则称这条道路为可增广道路。令

$$\delta_{ij} = \begin{cases} c_{ij} - f_{ij}, & \text{当}(i,j)\text{为向前边；} \\ f_{ij}, & \text{当}(i,j)\text{为后退边。} \end{cases}$$

$$\delta = \min\{\delta_{ij}\}。$$

所谓可增广道路也就是它的向前边均未饱和，向前边流量可增加，后退边流量可减少，后退边流量实为倒流量。即在这条道路上每条向前的边的流都可以提高一增量δ，而相应地后退的边的流减少δ，从而使得这个网络流的流量获得增加。同时保证使得每条边的流量不超过它的容量，而且保持为正的，也不影响其它的边的流量。总之，可增广道路的存在，便可以使得流量得到相应的增加。

现在我们证明定理：$\max w=\min C(S,\bar{S})$。

设网络流图G的网络流f使得流量达到极大。我们定义一割切$(S,\bar{S})$如下：

(a) $Z\in S$，

(b) 若$x\in S$，且$f_{xy}<c_{xy}$，则$y\in S$；

若$x\in S$且$f_{yx}>0$，则$y\in S$。

显然，收点$\bar{Z}\in\bar{S}=V\setminus S$；否则按子集$S$的定义存在一条从$Z$到$\bar{Z}$的道路：

$$Z = v_1, v_2, \cdots, v_k = \bar{Z}。$$

在这条道路上所有的向前边都满足：$f_{i,i+1}<c_{i,i+1}$，所有的向后边都满足$f_{i+1,i}>0$。因而这条道路是可增广道路，这和f是最大流的假设矛盾。因而$\bar{Z}\in\bar{S}$；即S和$\bar{S}$是个割切$(S,\bar{S})$。

按照子集 S 的定义，若 $x\in S$，$y\in\bar{S}$，则 $f_{xy}=c_{xy}$。若 $y\in\bar{S}$，$x\in S$，则 $f_{yx}=0$。

所以
$$w=\sum_{x\in S,\,y\in\bar{S}}[f_{xy}-f_{yx}]=\sum_{x\in S,\,y\in\bar{S}}c_{xy}=c(S,\bar{S})。$$

即
$$\max_f w=\min_f C(S,\bar{S})。\text{证毕}$$

从定理的证明中可以看到，利用逐渐增大流值的方法可以达到寻求最大流的目的，但这种方法实际做起来是有困难的。因为没有解决如何寻找可增广路的方法。下一节介绍的标号法将解决这个问题。

§4 标 号 法

在介绍标号法以前先证明 Minty 定理。

Minty 定理 对于连通图 G 的边任意着以黑、绿、红三种颜色，其中特定的 (Z,Z) 边着以黑色，则下面两种情况之一必然发生。

(*a*) 存在一条由红、黑色的边组成的回路 C，$e=(\bar{Z},Z)$ 是 C 上一条边，不仅如此，所有的黑色边的方向与回路 C 的方向一致。

(*b*) 存在一个包含 $e=(Z,\bar{Z})$ 边由黑、绿色边组成的割集 K，而且黑色边的方向与 e 边的方向一致。

证明 按下面的规则对各顶点加以标号。

边 $e=(\bar{Z},Z)$ 的终点 Z 给以标号。

对于一端点 x 已给标号，另一端点 y 未给标号的边，当 (x,y) 为黑色时，给 y 以标号，当 (x,y) 或 (y,x) 边是红色时，给 y 以标号。

按上面的规则执行直到最后。这样 $\bar{Z}$ 点可能给标号或不给标号两种可能，分别讨论如下

第一种情况 $\bar{Z}$ 得到了标号。则从 Z 出发存在一条和给标号过程相一致的回路
$$Z,\ v_1,\ v_2,\ \cdots,v_k,\ \bar{Z},\ Z$$
黑色边和这回路的方向一致。

第二种情况是 $\bar{Z}$ 得不到标号。令未标号的点集为 X，令以 X 的点为始点，终点不在 X 的边集合为 $K^+(X)$；以属于 X 的点为终点，始点不属于 X 的边集合为 $K^-(X)$。以 X 的点为一端点的边集合为 $K(X)$。由于假定 $\bar{Z}$ 未给标号，故
$$\bar{Z}\in X。$$
又根据假定 Z 给了标号，故边 $e=(\bar{Z},Z)\in K^-(X)$。根据给标号的规则，红色边不属于 $K(X)$；黑色边不属于 $K^-(X)$。这理由是显然的。由红色边只要一端点给了标号，另一端点也必然给了标号；黑色边不可能是始点有标号，而终点无标号。故 $K(X)$ 包含有绿色的边，以及始点在 X，终点不在 X 的黑色边。即黑色边的方向和 $(\bar{Z},Z)$ 方向一致。

后面介绍一种找最大流 f 的方法。它的基本思想是寻找一可增广道路，使网络流的流量得到增加，直到最大为止。标号法分为两个过程：一是标号过程，通过标号过程找到一条可增广道路；二是沿着可增广道路增加网络流流量的过程。

A：标记过程：

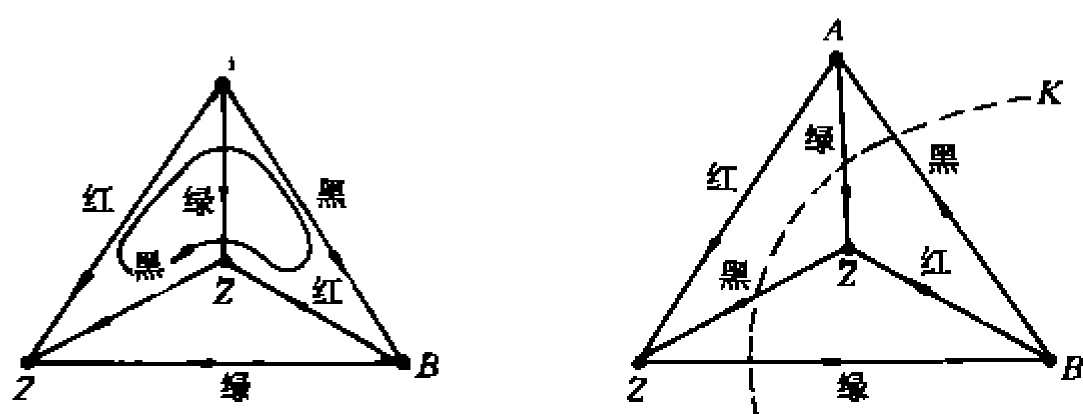


图 6-4-1

(1) 给发点 Z 以标记 $(+\bar{Z},\infty)$。

(2) 选择一个已给标记的顶点 x，对于 x 的所有未给标记的邻接点 y，按下列规则处理：

(a) 若边 $(y,x)\in E$，y 未给标记，而且 $f_{yx}>0$ 时，

令 $\delta_y=\min[f_{yx},\delta_x]$，则 y 给以标记 $(-x,\delta_y)$。

(b) 若 $(x,y)\in E$，y 未给标记，而且当 $c_{xy}>f_{xy}$ 时，

令 $\delta_y=\min[c_{xy}-f_{xy},\delta_x]$，则 y 给以标记 $(+x,\delta_y)$。

(3) 重复(2)直到收点 $\bar{Z}$ 被标记，或不再有顶点可以标记为止。如若 $\bar{Z}$ 点给了标记说明存在一条可增广道路，故转向增广过程 B。如若 $\bar{Z}$ 点不能获得标记，而且不存在其它可标记的顶点时，算法结束，所得到的流便是最大流。

B：增广过程：

(1)：令 $u=\bar{Z}$，

(2)：若 u 的标记为 $(+v,\delta)$，则

$$f_{vu}\leftarrow f_{vu}+\delta_{\bar{Z}};$$

若 u 的标记为 $(-v,\delta)$，则

$$f_{uv}\leftarrow f_{uv}-\delta_{\bar{Z}}。$$

(3) 若 $v=Z$，则把全部标记去掉，并转向标记过程 A。如若不然，令 $u=v$ 并回到增广过程的(2)。

若从 $\bar{Z}$ 到 Z 引一条虚拟的有向边 $e=(\bar{Z},Z)$，并给 Z 以标记。若遇到标号过程第二步的状态(a)时，边 (y,x) 着以红色。若遇到状态(b)时，边 (x,y) 着以黑色。其余的边属于向前的饱和边，或流量为 0 的后退边时，着以绿色。根据 Minty 三色引理可知：或存在包含 $(\bar{Z},Z)$ 边，由红、黑色边组成的回路；或存在包含 $(\bar{Z},Z)$ 边，由黑、绿色边组成的割切。前者为存在可增广道路，后者有最大流。

下面我们通过例子(见图 6-4-2(a))说明标号法如下。图中各边都标以一对有序数，第一个数是该边容量，第二个数是该边的流量。

从各边的流量为零开始，标记法的全部过程如图 6-4-2(b)所示。

从上面的标记过程可知，当一顶点 v 被标记时，说明从发点 Z 到 v 点的流可以增加 δ_v。如若 $\bar{Z}$ 被标记表明从 Z 到 $\bar{Z}$ 存在一条可增广道路，这条道路上的流的增量由 $\delta_{\bar{Z}}$ 确定。

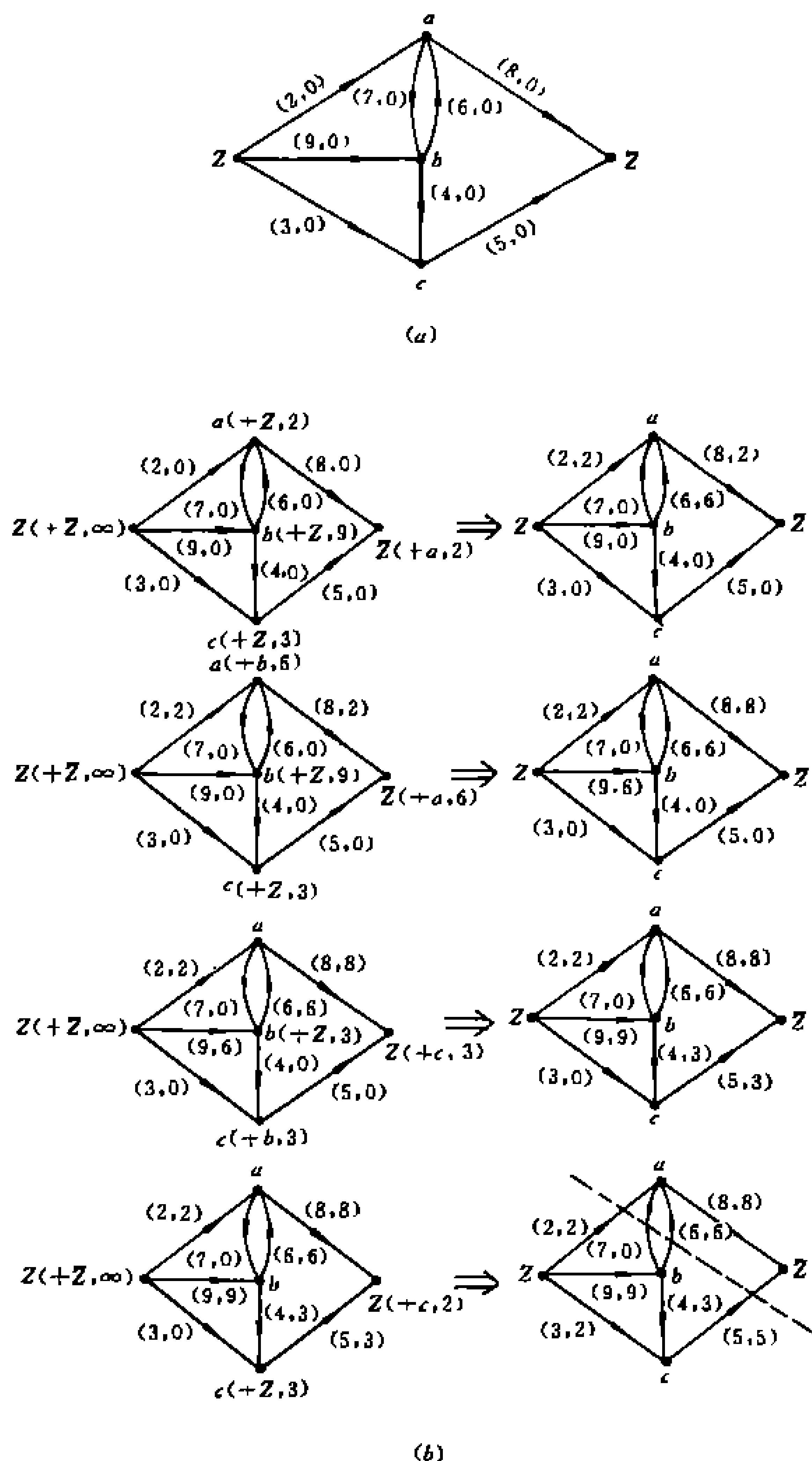


图　6-1-2

当边的容量都是正整数时，对每一次增广过程，至少使网络流的流量增加一个单位。由于极大流也是正整数，故可在有限步骤内使网络流达到极大。类似的理由可说明：当各边的流量为有理数时，可在有限步骤内使网络流达到最大。

§5 Edmonds-Karp 修正算法，Dinic 算法及其它

(1) Edmonds-Karp 修正算法

Ford-Fulkerson 算法理论上存在着严重的弱点，以下面图 6-5-1 为例各边上的权是它们的容量，若交替地采用 $Zab\bar{Z}$ 和 $Zba\bar{Z}$ 作为增广道路，当初始流量 f_0 为零时，无疑需作 $2m$ 次的增加流量才能使之达到最大，可见 Ford-Fulkerson 算法的时间复杂度不仅依赖于网络的规模（即依赖于网络点数和边数），还和各边的容量有关。以图 6-5-1 为例，当 $Zab\bar{Z}$ 和 $Zba\bar{Z}$ 交替作为增广道路时，ab 边交替地用向前边和后退边出现。Ford-Fulkerson 算法的复杂性分析变得很困难了。

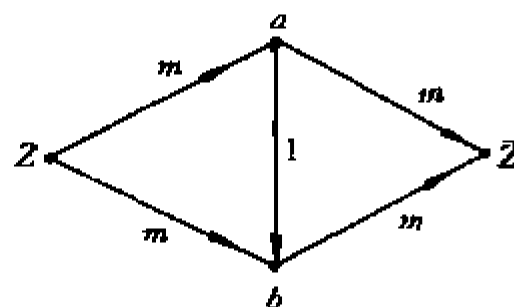


图 6-5-1

Edmonds 和 Karp 对 Ford-Fulkerson 法作修正，可概括为一句话："先给标记的先扫描。"它的意思是对已给标记的顶点 v 进行扫描时，先对所有和 v 邻接的未给标记的顶点给予标记。具体的说图 6-5-1 的例子，顶点 Z 先标记，所以应该先扫描。因此避免了 Ford-Fulkerson 算法那样交替地出现 $Zab\bar{Z}$，$Zba\bar{Z}$ 的情况；也就避免了 ab 边交替地以向前边，和后退边来回摇摆的局面。按照 Edmonds-Karp 算法，Z 首先标记，对 Z 扫描时先后对 a 和 b 标记；进而对 a 扫描，所以 Edmonds-Karp 的修正实质是对顶点给标记过程采用了"宽度优先"策略。使得流量增加总是沿着一条长度最短的道路从 Z 流向 $\bar{Z}$ 的。

假定 f_0 是网络的初始流，通过 Edmonds-Karp 标记法依次得流序列：$f_0, f_1, \cdots, f_k, f_{k+1}, \cdots$

设 f_{k+1} 的一条增广道路是

$$P: Z = u_0 \xrightarrow{e_1} u_1 \xrightarrow{e_2} u_2 \cdots\cdots u_{l-1} \xrightarrow{e_l} u_l = \bar{Z}$$

令

$$\varepsilon_i^P = \begin{cases} c(e_i) - f_k(e_i), & e_i \text{ 是向前边}, \\ f(e_i), & e_i \text{ 是后退边}。\end{cases}$$

$$\varepsilon^P = \min_i \{\varepsilon_i^P\} = \varepsilon_j^P,$$

其中 $c(e_i)$ 为 e_i 边的容量，$f_k(e_i)$ 为 e_i 边的流量，$\min\{\varepsilon_i^P\} = \varepsilon_j^P$ 说明 e_j 边是"瓶颈"。通过标记法使得流 $f(e_j)$ 产生如下的变化，若 e_j 边是向前边，则令 $c(e_j) = f_{k+1}(e_j)$，即使 e_j 边达到饱和；若 e_j 边是后退边，则令 $f_{k+1}(e_j) = 0$，即使之倒流量降为零。

令 $l_i(u,v)$ 表示流 f_i 中由 u 到 v 最短未饱和路径的长度。既然是未饱和路径，路径上的向前边 e_j 必然有 $c(e_j) > f_i(e_j)$，后退边 e_l 有 $f_i(e_l) > 0$。

假定下面是一条由 Z 到 u 流 f_k 最短的未饱和的路径

$$Z = u_0 \xrightarrow{e_1} u_1 \xrightarrow{e_2} u_2 \cdots u_{l-1} \xrightarrow{e_l} u_l = u$$

若 e_j 是这路径上一条向前边，则有

$$c(e_j) > f_k(e_j)。$$

对于 f_{k+1} 自然有

$$c(e_j) \geqslant f_{k+1}(e_j)。$$

若 e_j 是后退边则有 $0<f_k(e_j)\leqslant c(e_j)$，而且

$$0 \leqslant f_{k+1}(e_j) \leqslant c(e_j)。$$

也就是说从 Z 到 u，f_k 流未饱和边的最短路径上有可能出现 f_{k+1} 流的饱和边，故有

$$l_k(Z,u) \leqslant l_{k+1}(Z,u)。$$

类似的理由有

$$l_k(u,\bar{Z}) \leqslant l_{k+1}(u,\bar{Z})。$$

下面这个结论是很重要的：

引理 利用 Edmonds-Karp 修正算法，$e=(u,v)$ 作为增广路径中的向前边从 f_k 流转变为 f_{k+1} 流，但在 f_h 流中作为增广路径中的后退边转变为 f_{h+1}，$h>k$，则

$$l_h(Z,\bar{Z}) \geqslant l_k(Z,\bar{Z}) + 2。$$

也就是从 Z 到 $\bar{Z}$ 的最短路径 f_h 流比 f_k 流至少要长出 2。由于 $l_h(Z,u)\geqslant l_k(Z,u)$，$l_h(v,\bar{Z})\geqslant l_k(v,\bar{Z})$，

$$l_k(Z,\bar{Z}) = l_k(Z,u) + l_k(v,\bar{Z}) + 1,\ l_k(Z,v) = l_k(Z,u) + 1,$$

$$l_h(Z,\bar{Z}) = l_h(Z,v) + l_h(u,\bar{Z}) + 1 \geqslant l_k(Z,v) + l_k(u,\bar{Z}) + 1 = l_k(Z,u) + l_k(u,\bar{Z}) + 2,$$

$$\therefore \quad l_h(Z,\bar{Z}) \geqslant l_k(Z,\bar{Z})+2。$$

请注意这里的 $l_h(Z,v)$，$l_h(u,\bar{Z})$ 等都是利用 Edmonds-Karp 算法所得的最短距离。

下面对 Edmonds-Karp 算法的复杂性进行估计。

假定网络图有 n 个顶点，m 条边。则从 Z 到 $\bar{Z}$ 的最长路径不超过 $n-1$。每条边最多作为增广路径上的后退边反复了 $\frac{1}{2}(n-1)$ 次，所以增广路径数不超过

$$\frac{1}{2}(n-1)m$$

也就是 Edmonds-Karp 算法在不超过 $\frac{1}{2}(n-1)m$ 次增广过程达到最大流。

(2) Dinic 算法

Dinic 算法的特点是将顶点按其与 Z 点的最短距离分层。Edmonds-Karp 算法的实质也是一种分层。如果说 Ford-Fulkerson 算法是采用了深度优先策略，Edmonds-Karp 算法则是将宽度优先取代了深度优先。Dinic 算法则是兼取这两种方法，在分层时用的宽度优先法，而寻求增广路径时则采用深度优先策略。

Dinic 算法

(1) 对所有 $e\in E$，$f(e)\leftarrow 0$。

(2) $L_0\leftarrow\{Z\}$，$i\leftarrow 0$，$R\leftarrow V\setminus\{Z\}$。

(3) $S\leftarrow\{v \mid v\in R$，且存在一条从 L_i 某一顶点到 v 的未饱和边$\}$。

(4) 若 $S=\varnothing$，则网络流已达到最大，停止。否则若 $\bar{Z}\notin S$ 则转(5)，否则转(6)。

(5) $i\leftarrow i+1$，$L_i\leftarrow S$，$R\leftarrow R\setminus\{L_i\}$，转(3)。

(6) $u\leftarrow Z$，$T\leftarrow\varnothing$。

(7) 选择未饱和边 $e=(u,v)$，其中 v 在后面一层，将 e 进入栈 T，$u\leftarrow v$，若 $u\neq t$ 则转(7)，否则转(8)。

(8) 存在一条从 Z 到 $\bar{Z}$ 的增广路径 P 作

$$\delta(e)=\begin{cases}c(e)-f(e), & \text{若 } e \text{ 是向前边}, \\ f(e), & \text{若 } e \text{ 是后退边},\end{cases}$$

$\forall e\in P$。

$\delta=\min_{e}\{\delta(e)\}$。

(9) 对所有 $e\in P$ 作

$$f(e)=\begin{cases}f(e)+\delta, & \text{若 } e \text{ 是向前边}, \\ f(e)-\delta, & \text{若 } e \text{ 是后退边}。\end{cases}$$

(10) 转(2)

Dinic 算法的时间复杂性为 $O(n^2m)$，现通过例子说明如下。

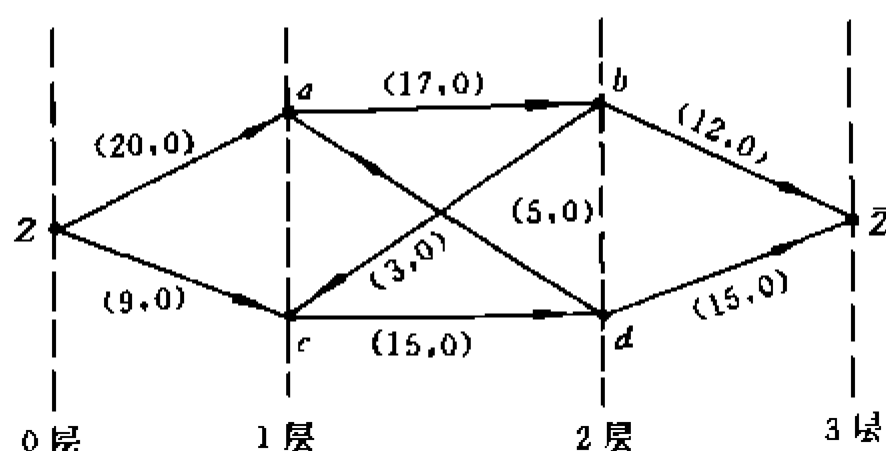


图 6-5-2

边上( )里第一个数是边容量，第 2 个数是该边的流 $f(e)$。

第一次应用 Dinic 算法得图 6-5-3：

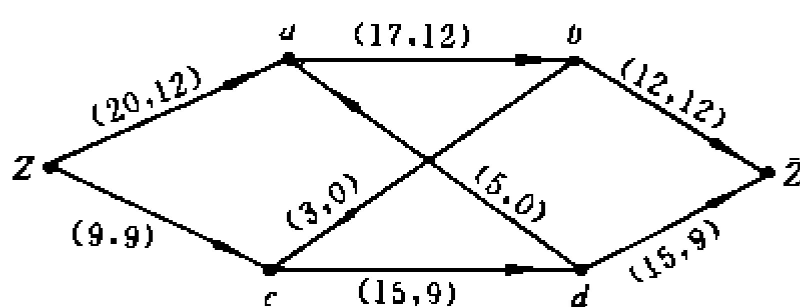


图 6-5-3

流量为 $12+9=21$。

根据非饱和边构成的导出网络见图 6-5-4，对图 6-5-4 进行分层得图 6-5-5：

再利用分层法得图 6-5-6。

继续下去得 $S=\varnothing$。故得最大流见图 6-5-7。

(3) 最后，我们简单地谈一谈最大流最小割切定理的推广。

(a) 多产地多销地问题：

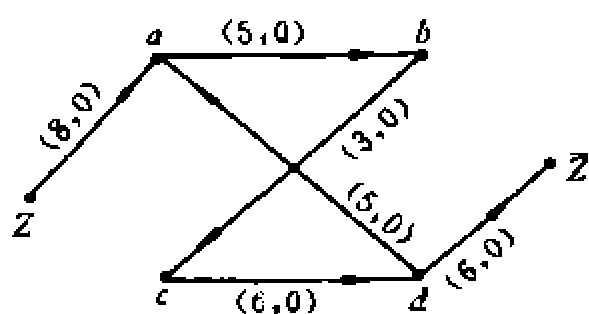


图 6-5-4

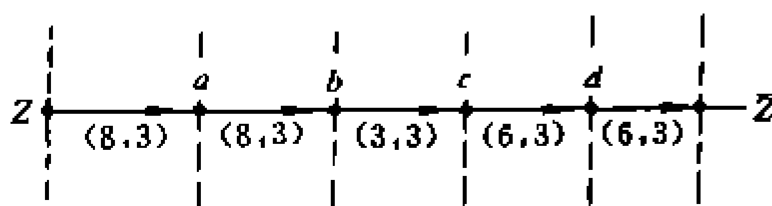


图 6-5-5

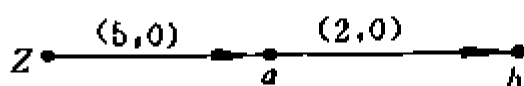


图 6-5-6

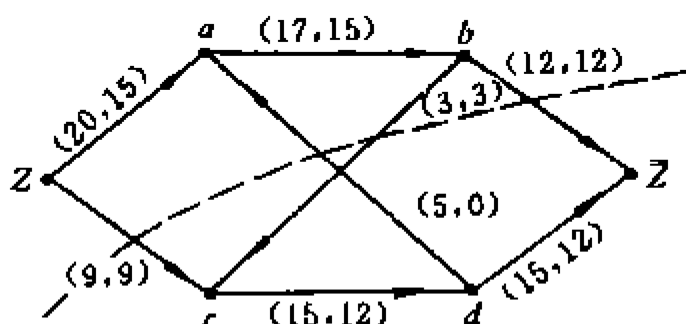


图 6-5-7

某一种物资若有 l 个发点 $Z_1,Z_2,\cdots,Z_l$，m 个收点 $\bar{Z}_1,\bar{Z}_2,\cdots,\bar{Z}_m$。如图 6-5-8 所示，对于这种网络，我们放人两个顶点 Z 和 $\bar{Z}$，Z 作为新的发点，$\bar{Z}$ 作为新的收点，连接新的弧 $(Z,Z_1),(Z,Z_2)\cdots,(Z,Z_l)$ 和 $(\bar{Z}_1,\bar{Z}),(\bar{Z}_2,\bar{Z})\cdots,(\bar{Z}_m,\bar{Z})$ 指定它们的容量均为 $+\infty$，这样就把原来的网络化为一个发点和一个收点的网络。在求得最大流后，通过 Z_i 的物资均看作是由 Z_i 发点，经由 $\bar{Z}_i$ 的物资均由 $\bar{Z}_i$ 收留。

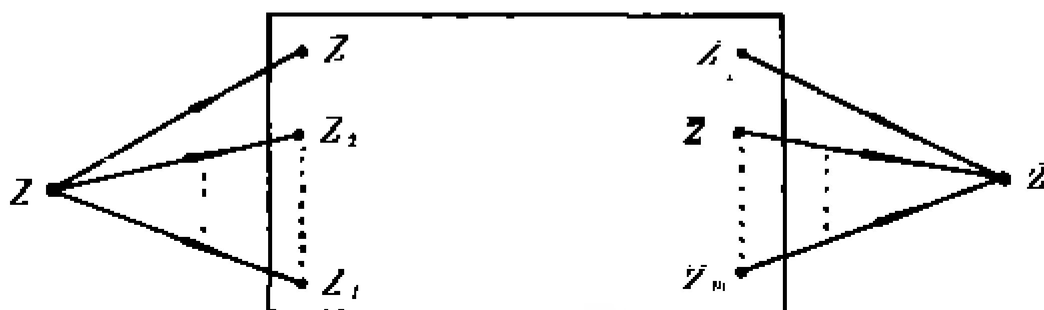


图 6-5-8

(b) 顶点具有容量的网络：

假如在一个网络上不仅弧具有容量而且顶点也具有容量，顶点的容量受到限制，而弧上的容量不受限制。对于这种类型的网络不难把它变为我们已经讨论过的网络，具体过程是：把每个顶点 v_i 分为两个顶点 $v_i^{(1)}$ 和 $v_i^{(2)}$，同时在 $v_i^{(1)}$ 和 $v_i^{(2)}$ 之间连一条弧 $(v_i^{(1)},v_i^{(2)})$，并且约定所有可达 v_i 的顶点都改为到达 $v_i^{(1)}$；而把所有的弧 (v_i,v_j) 改为 $(v_i^{(2)},v_j)$，并给弧 $(v_i^{(1)},v_i^{(2)})$ 以容量 $c(v_i^{(1)},v_i^{(2)})=c(v_i)$，而将原有各弧均令其容量为 $+\infty$。这样，就可以把顶点也具有容量的网络上的最大流问题化为我们已经讨论过的网络上的问题了。

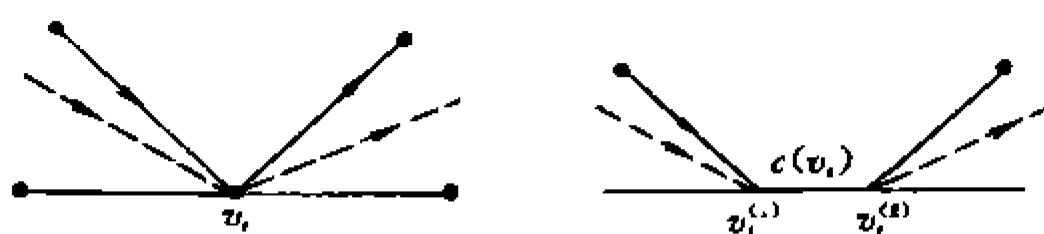


图 6-5-9

§6 开关网络简介

开关网络是计算机设计中的重要课题，在其它通讯系统方面也有应用。可以把开关网络看作是一无向的连通图，图的每一条边都对应有某一布尔变量 x_i 作为该边的权。x_i 可以看作是该边上的接触开关，当开关接通时 x_i 取值 1，否则 x_i 取值 0。这样的开关网络用 G_N 表示它。

设 a、b 是开关网络 G_N 上两个顶点，而 $P_{ab}^{(k)}$ 是 a、b 两点间的道路，其中 $k=1,2,\cdots,n$。若 $P_{ab}^{(k)}$ 道路上各边的权的连乘积为 $\pi_{ab}^{(k)}$，并令：

$$f_{ab}=\sum_{k=1}^{l}\pi_{ab}^{(k)}。$$

称 f_{ab} 为开关网络 G_N 关于顶点 a,b 的开关函数。例：图 6-6-1 中 a,b 间的道路有：$x_1x_3x_7$，$x_2x_4x_8$，$x_1x_5x_8$，$x_2x_6x_7$，$x_1x_3x_6x_4x_8$，$x_2x_4x_5x_3x_7$，$x_2x_6x_3x_5x_8$，$x_1x_5x_4x_6x_7$；故有

$$\begin{aligned}f_{ab}= & x_1x_3x_7+x_2x_4x_8+x_1x_5x_8+x_2x_6x_7\\ & +x_1x_3x_6x_4x_8+x_2x_4x_5x_3x_7\\ & +x_2x_6x_3x_5x_8+x_1x_5x_4x_6x_7。\end{aligned}$$

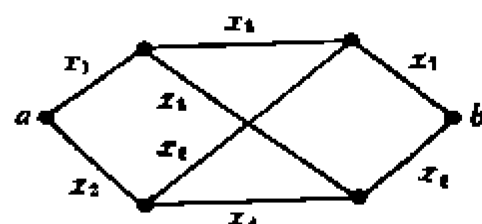


图 6-6-1

上式中的乘积为逻辑乘，和为逻辑和，故服从逻辑运算规则：

$$1+x=1,\quad x+\bar{x}=1,\quad xx=0,$$
$$x+x=xx=x+xy=x。$$

其中布尔变量 $x_1,x_2,x_3,\cdots,x_8$ 可以是独立的变量，也可以是有相同的。比如若：

$$x_1=x,\quad x_2=x,\quad x_3=y,\quad x_4=y,$$
$$x_5=y,\quad x_6=y,\quad x_7=\bar{Z},\quad x_8=Z。$$

则 $f_{ab}=x\bar{y}\bar{Z}+xyZ+xyZ+\bar{x}y\bar{Z}+xyyyZ+xyy\bar{y}\bar{Z}+xyyy\bar{Z}+xyyyZ$。

由布尔量的运算规则，上述开关函数可以简化为：

$$f_{ab}=x\bar{y}\bar{Z}+\bar{x}\bar{y}Z+xyZ+\bar{x}y\bar{Z}。$$

如果开关网络 G_N 的所有的边的权都不相同时，称为是简单接触的网络。故简单接触网络中的开关都是独立的，即可以独立地接通或断开。

例如：

图 6-6-2(a)是简单接触网络，而(b)则不是，(a)的开关函数 f_{ab} 为：

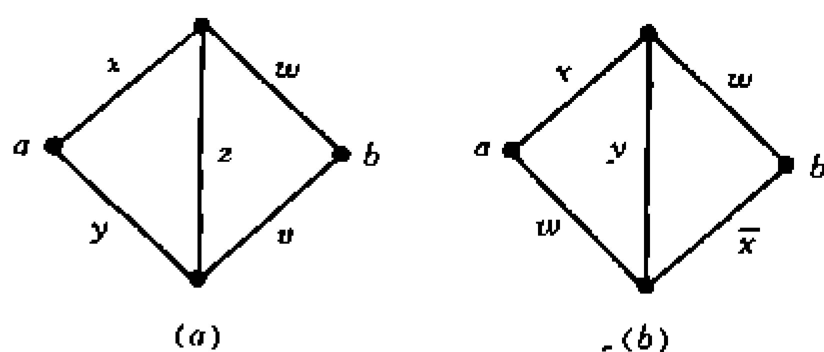


图 6-6-2

$$f_{ab} = xw + yv + xzv + yzw_{\circ}$$

而(b)的开关函数f_{ab}为：

$$\begin{aligned} f_{ab} &= xw + \bar{x}w + xy\bar{x} + wyw \\ &= xw + \bar{x}w + yw \\ &= w(x + \bar{x}) + yw \\ &= w_{\circ} \end{aligned}$$

说明x和y的开关是多余的，可以省去。

对于开关网络$G_N = (V,E)$，有矩阵

$$\boldsymbol{F} = (f_{ij})_{n\times n},$$

其中

$$f_{ij} = \begin{cases} 1, & i = j, \\ \text{顶点 } i、j \text{ 间的开关函数}, & i \neq j_{\circ} \end{cases}$$

$$i, j = 1, 2, \cdots, n, \quad n = |V|_{\circ}$$

则称矩阵F为开关网络的传输矩阵。

如果说连接矩阵A类似于邻接矩阵，而传输矩阵颇与道路矩阵相当。不难得到关系式：

$$\boldsymbol{F} = \boldsymbol{A}^{(n-1)}_{\circ}$$

这里$\boldsymbol{A}^{(n-1)}$是矩阵$\boldsymbol{A}$的$n-1$次幂，不过乘是逻辑乘，和是逻辑和，并服从逻辑运算法则。

例：

$$\boldsymbol{A} = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}\begin{pmatrix} 1 & x_1 & x_2 & 0 \\ x_1 & 1 & x_3 & 0 \\ x_2 & x_3 & 1 & x_4 \\ 0 & 0 & x_4 & 1 \end{pmatrix},$$

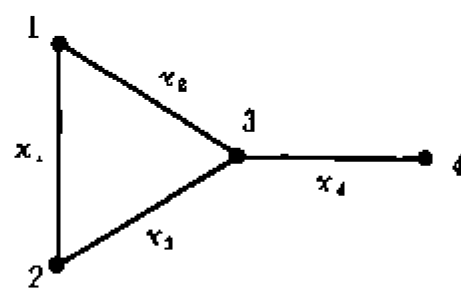


图 6-6-3

$$\boldsymbol{A}^{(2)} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ & \begin{pmatrix} 1 & x_1 & x_2 & 0 \\ x_1 & 1 & x_3 & 0 \\ x_2 & x_3 & 1 & x_4 \\ 0 & 0 & x_4 & 1 \end{pmatrix} \end{matrix} \begin{pmatrix} 1 & x_1 & x_2 & 0 \\ x_1 & 1 & x_3 & 0 \\ x_2 & x_3 & 1 & x_4 \\ 0 & 0 & x_4 & 1 \end{pmatrix}$$

$$
=\begin{pmatrix}1+x_1x_1+x_2x_2 & x_1+x_1+x_2x_3 & x_2+x_1x_3+x_2 & x_2x_4\\ x_1+x_1+x_2x_3 & x_1x_1+1+x_3x_3 & x_1x_2+x_3+x_3 & x_3x_4\\ x_2+x_1x_3+x_2 & x_1x_2+x_3+x_3 & x_2x_2+x_3x_3+1+x_4x_4 & x_4+x_4\\ x_2x_4 & x_3x_4 & x_4+x_4 & 1+x_4x_4\end{pmatrix}
$$

$$
=\begin{pmatrix}1 & x_1+x_2x_3 & x_2+x_1x_3 & x_2x_4\\ x_1+x_2x_3 & 1 & x_3+x_1x_2 & x_3x_4\\ x_2+x_1x_3 & x_3+x_1x_2 & 1 & x_4\\ x_2x_4 & x_3x_4 & x_4 & 1\end{pmatrix}。
$$

请注意 $a_{ij}^{(2)}$ 实际上是顶点 i,j 间"不超过两步"走到的道路的开关函数。

$$
A^{(3)}=\begin{pmatrix}1 & x_1+x_2x_3 & x_2+x_1x_3 & x_2x_4\\ x_1+x_2x_3 & 1 & x_3+x_1x_2 & x_3x_4\\ x_2+x_1x_3 & x_3+x_1x_2 & 1 & x_4\\ x_2x_4 & x_3x_4 & x_4 & 1\end{pmatrix}\begin{pmatrix}1 & x_1 & x_2 & 0\\ x_1 & 1 & x_3 & 0\\ x_2 & x_3 & 1 & x_4\\ 0 & 0 & x_4 & 1\end{pmatrix}
$$

$$
=\begin{pmatrix}1 & x_1+x_2x_3 & x_2+x_1x_3 & x_4(x_2+x_1x_3)\\ x_2+x_1x_3 & 1 & x_3+x_1x_2 & x_4(x_3+x_1x_2)\\ x_2+x_1x_3 & x_3+x_1x_2 & 1 & x_4\\ x_4(x_2+x_1x_3) & x_4(x_3+x_1x_2) & x_4 & 1\end{pmatrix}。
$$

习　　题

1. 用标号法求下图运输网络的最大流。其中无方向的边是双向的。

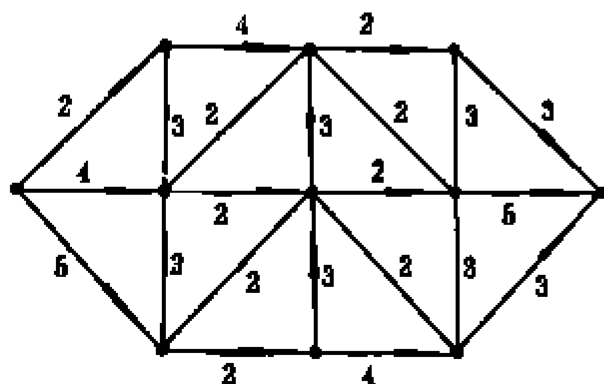


图　习题 1

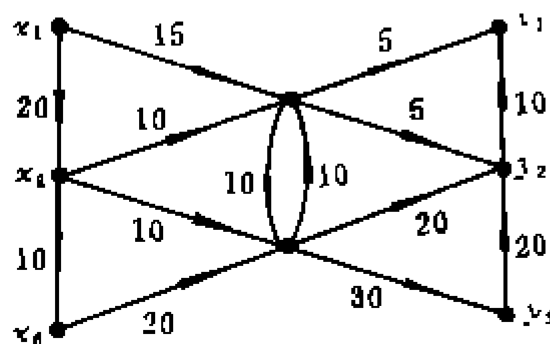


图　习题 2

2. 设 x_1,x_2,x_3 是三家工厂，y_1,y_2,y_3 是三个仓库。工厂生产的产品要运往仓库，其运输网络如图所示。设 x_1,x_2,x_3 的生产能力分别为 40，20，10 个单位。问应如何安排生产？
3. 七种设备要用五架飞机运往目的地，每种设备各有 4 台，这五架飞机容量分别是 8、8、5、4、4 台，问能否有一种装法使同一种类型设备不会有两台在同一架飞机上？
4. 上题中若飞机的容量分别是 7、7、6、4、4 台，求问题的解。
5. 用 Edmonds Karp 修正算法求问题 1 的解。
6. 用 Dinic 算法求问题 1 的解。
7. 写出开关函数 $f=x_1\bar{x}_2+x_2\bar{x}_1$ 的真值表，并利用它设计一开关电路，使得楼上楼下都能

开启或关闭一盏电灯。

8. 求下图网络的最大流，边上的数是边容量。

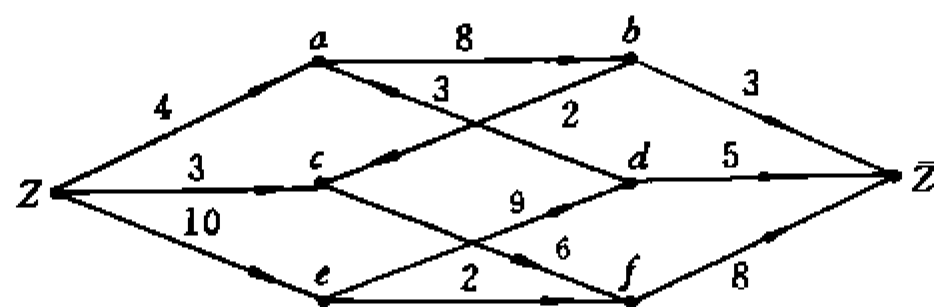


图　习题 8

9. 试用 Edmonds-Karp 和 Dinic 算法求问题 8 的解。

第七章　匹配理论、色数问题及其它

本章讨论匹配，色数等一些运筹学中的问题。

“匹配”是图论中的一个重要内容，它在所谓“人员分配问题”和“最优分配问题”中有重要应用。

§1　最大匹配

我们首先来看实际中常碰见的一个问题：给 n 个工作人员安排 m 项任务。n 个人用 $X(x_1,x_2,x_3,\cdots,x_n)$ 表示，而 m 项任务用 $Y(y_1,y_2,y_3,\cdots,y_m)$ 表示。并不是所有的工作人员都能从事任何一项工作，比如 x_1 能做 y_1,y_2 工作，而 x_2 可以做 y_2,y_3 工作等等。问应如何安排工作才能做到最大限度地使任务有人去做，使得更多的人有工作做。

如图 7-1-1，x_1,x_2,x_3,x_4,x_5 为五个人，y_1,y_2,y_3,y_4,y_5 为五项任务。x_1 能做 y_1,y_2 两项工作，则联 x_1y_1,x_1y_2，如图 7-1-1 所示，x_2 能做 y_2,y_3 两项工作，x_3 能做 y_2,y_5 两项工作，x_4 只能做 y_3 一项工作，x_5 可从事 y_3,y_4,y_5 三项工作。如何安排工作尽可能使每个人能从事一项工作，每项工作也由一人作。

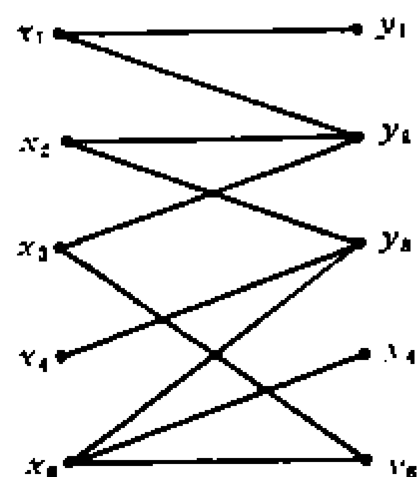


图　7-1-1

所谓匹配问题就是要从图 G 中找一边的子集，使得每一个顶点最多只能与一条边相关联。

下面我们首先介绍几个基本概念：

(1) 二分图(偶图)：

若图 $G=(V,E)$ 的顶点集合 V 可以分成两个满足下述条件的子集 X、Y：

① $X\cup Y=V$，$X\cap Y=\varnothing$。

② 其邻接矩阵具有如下的形式：

$$A=\begin{matrix} X \\ Y \end{matrix}\begin{pmatrix} O & A_{12} \\ A_{21} & O \end{pmatrix}$$
$$\qquad\quad X \qquad Y$$

则称图 G 是二分图(或偶图)。

(2) 匹配

设 M 是 $E(G)$ 的一个子集，如果 M 中任意两条边在 G 中均不邻接，则称 M 是 G 的一个匹配。M 中的一条边的两个端点叫做在 M 是配对的。

(3) 饱和与非饱和：

若匹配 M 的某条边与顶点 v 关联，则称 M 饱和顶点 v，并且称 v 是 M－饱和的，否则

称 v 是 M—不饱和的。

（4）交互道：

若 M 是二分图 $G=(V,E)$ 的一个匹配。设从图 G 中的一个顶点到另一个顶点存在一条道路，这条道路是由属于 M 的边和不属于 M 的边交替出现组成的，则称这条道路为交互道。

如图 7-1-2 中 $x_1y_1x_3y_2x_2y_3x_4y_4$ 便是一条从 x_1 到 y_4 的交互道。图中用实线表示属于匹配 M 的边。虚线则不是。

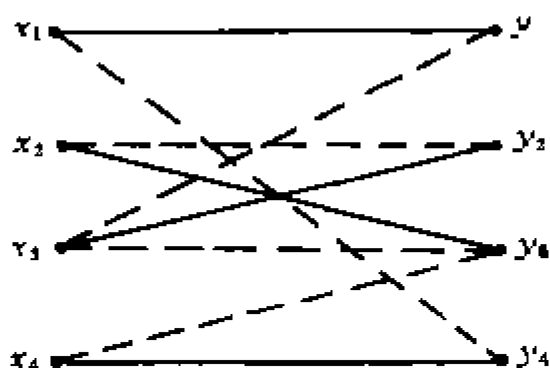


图 7-1-2

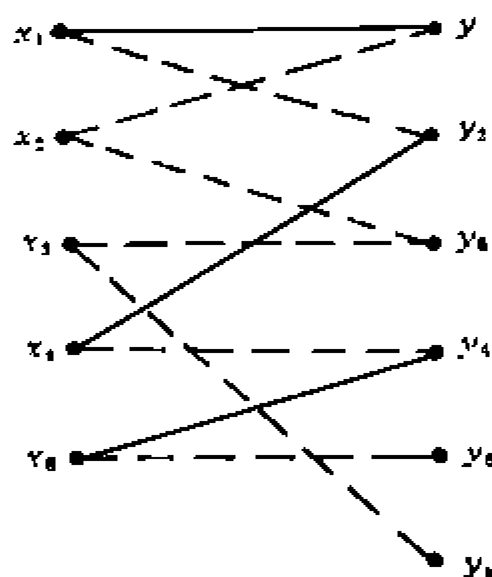


图 7-1-3

（5）可增广道路：

若一交互道的两端点为关于 M 非饱和顶点时，则称这条交互道是可增广道路。显然，一条边的两端点非饱和，则这条边也是可增广道路。

例如图 7-1-3 中 $x_2y_1x_1y_2x_4y_4x_5y_5$ 也是一条交互道，但其两端点 x_2，y_5 都是非饱和的，故是一条可增广道路；当然 x_3y_6 也是一条可增广道路。

其中由于可增广道路的两端点是非饱和顶点，非 M 的边和 M 的边交互出现，故可增广道路的边数必为奇数，非 M 的边比 M 的边多一条。对于可增广道路只要改变一下匹配关系，即只要把这道路上非 M 的边改为属于 M 的边，而属于 M 的边改为非 M 的边，这样可以使得匹配的边数增加一条。如图 7-1-3 中的可增广道路，可按上述的办法提高匹配的边数如图 7-1-4 所示。结果使得匹配的边从 3 增到 5。

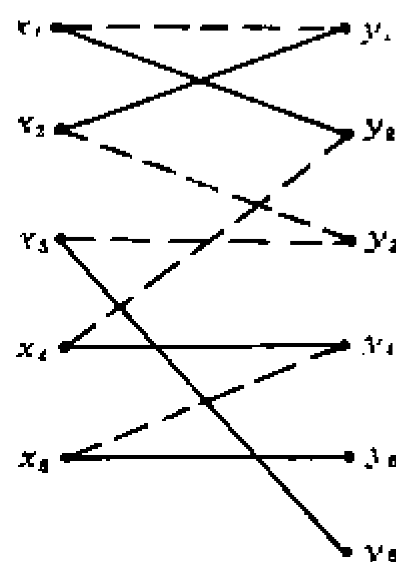


图 7-1-4

（6）最大匹配：

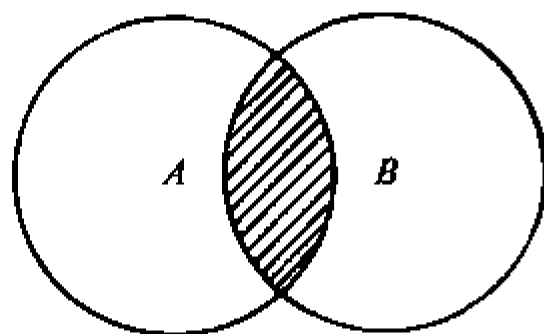
如果 M 是一匹配，而不存在其它匹配 M'，使得 $|M'|>|M|$，则称 M 是最大匹配。其中 $|M|$ 表示匹配 M 的边数。

（7）对称差：

A，B 是两个集合，定义：

$$A \oplus B \triangleq (A \cup B) \setminus (A \cap B)。$$

如图 7-1-5 所示，影线部分是 $A \cap B$，无影线部分为 $A \oplus B$。运算规则：$0 \oplus 0 = 1 \oplus 1 = 0$，$1 \oplus 0 = 0 \oplus 1 = 1$。



$A \oplus B$

图 7-1-5

定理 M 为最大匹配的充要条件是不存在可增广道路。

证明 必要性：设 M 是最大匹配，证不存在可增广道路。

用反证法。设 M 是最大匹配，并且存在一条可增广道路（这条道路的边数为奇数）P：

$$v_0v_1v_2\cdots v_{2m}v_{2m+1}。$$

其中 $v_0v_1, v_2v_3, v_4v_5, \cdots, v_{2m}v_{2m+1}$ 为非匹配的边，而 $v_1v_2, v_3v_4, \cdots, v_{2m-1}v_{2m}$ 为匹配边。令 $E(P)$ 表示可增广道路 P 的边的集合：

$$E(P) = \{(v_0,v_1),(v_1,v_2),\cdots,(v_{2m},v_{2m+1})\}。$$

则令

$$M_1 = M \oplus E(P) = (M \cup E(P)) \setminus (M \cap E(P))$$

$$= (M \cup \{(v_0,v_1),(v_1,v_2),\cdots,(v_{2m},v_{2m+1})\}) \setminus \{(v_1,v_2),(v_3,v_4),\cdots,(v_{2m-1},v_{2m})\}。$$

集合 M_1 本身也是一种匹配，而 $|M_1| = |M|+1 > |M|$，与 M 是最大匹配的假定矛盾。必要性证毕。

充分性：即若不存在关于 M 的可增广道路。则 M 是最大匹配。

仍用反证法。如果 M 不是最大匹配，则一定存在一匹配 M_1，$|M_1| > |M|$。作 $M_2 = M_1 \oplus M$。

(1) M_2 中的各连通部分必定是关于 M 和 M_1 的交互道。因为 M_2 是由 M_1 和 M 中非共同部份组成的，而 M 和 M_1 都是匹配，各自边之间没有共同的顶点，所以任何连通的道路都必然是交替出现 M_1 和 M 的边。

(2) 由于 $|M_1| > |M|$，所以在所有的交互道中，必有这样一条交互道，属于 M_1 的边多于属于 M 的边，这条道路必然是以 M_1 的边开始，且以 M_1 的边结束的交互道，即为关于 M 的可增广道路。与 M 是最大匹配的假设矛盾。充分性得到证明。

实际上这定理的证明过程本身，已为我们提供一条找最大匹配的方法：先给任一匹配，从中找出一条可增广道路，然后修改匹配，这样反复进行直到不存在可增广道路为止。

§2 Hall 定 理

设有 m 个人，n 项任务，能不能适当地安排，使得每个人都有工作做？当 $n<m$ 时，答案显然是否定的，即使 $n \geqslant m$ 也不一定。但经验告诉我们，当 m 个人适应工作的能力越强时，越容易做到这一点。Hall 定理则定量地回答了这个问题：

定理 对于二分图 G，存在一匹配 M，使得 X 的所有顶点关于 M 饱和的充要条件是：对于 X 的任一子集 A，和 A 邻接的点集为 $\Gamma(A)$，恒有：

$$|\Gamma(A)| \geqslant |A|。$$

证明

必要性：若存在一个匹配 M，使得 X 关于 M 饱和 $|\Gamma(A)| \geqslant |A|$ 的直观意义是显然的（即不论多少人，他们能做的工作数目不少于人数，这样定能做到每个人至少有一项彼此不同的工作）。

充分性：若对于任何 $A \subseteq X$，恒有：

$$|\Gamma(A)| \geqslant |A|。$$

则可以按下列方法作出匹配 M，使得 X 关于 M 饱和。先作任意一初始匹配，若已使 X 饱和，则定理已证。如若不然，则 X 中至少有一点 x_0 非饱和，则从 x_0 出发，检查从 x_0 出发，终点在 Y 的交互道。可能有下列两种情况发生：

（1）没有任何一条交互道可以到达 Y 的非饱和点。这时由于从 x_0 点开始的一切交互道终点还是在 X，故存在 X 的子集 A 有：

$$|\Gamma(A)| < |A|。$$

这与假设相矛盾，所以这种情形是不可能的。

（2）存在一条从 x_0 出发的交互道，终点为 Y 的非饱和点，则这条道路便是可增广道路，因而可以改变一下匹配使 x_0 点饱和。

重复以上的过程，就可以找出匹配 M，使 X 全部饱和。定理的充分性得到证明。

§3　匈牙利算法及例

匈牙利算法是求二分图最大匹配的一种算法，它的根据就是 Hall 定理中充分性证明中的思想，现将算法叙述如下：

（1）任给初始匹配。

（2）若 X 已饱和则结束，否则转（3）。

（3）在 X 中找一非饱和点 x_0，作

$$V_1 \leftarrow \{x_0\}, \qquad V_2 \leftarrow \varnothing$$

（4）若 $\Gamma(V_1) = V_2$ 则因无法匹配而停止；否则任选一点 $y \in \Gamma(V_1) \backslash V_2$。

（5）若 y 已饱和转（6），否则作

【求一条从 $x_0 \rightarrow y$ 的可增广道路 P，$M \leftarrow M \oplus E(P)$，转（2），】。

（6）由于 y 已饱和，故 M 中有一条边 (y, z) 作

【$V_1 \leftarrow V_1 \cup \{z\}$，$V_2 \leftarrow V_2 \cup \{y\}$，转（4），】。

现将算法的流程图表示见图 7-3-1。

例求图 7-3-2 的匹配 M，使 X 饱和。

第一步：给初始匹配 $M = \{x_1y_1, x_3y_5, x_5y_3\}$。如图 7-3-3 所示，属于匹配 M 的边用实线，其余用虚线。

第二步：显然 X 尚未饱和，找出其中一未饱和点 x_2，从 x_2 出发经过下列过程：

V_1：$\{x_2\} \rightarrow \{x_2, x_5\}, \rightarrow \{x_2, x_5, x_3\}$。

V_2：$\varnothing \rightarrow \{y_3\} \rightarrow \{y_3, y_5\} \rightarrow \{y_3, y_5, y_2\}$。

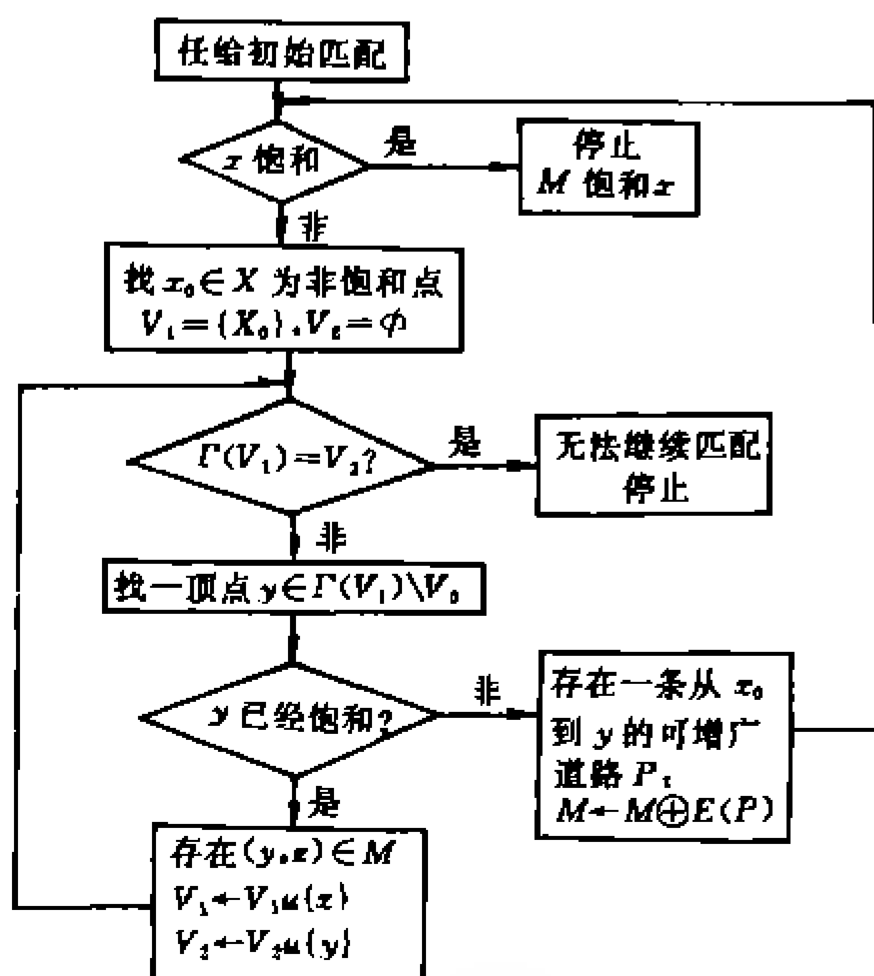


图 7-3

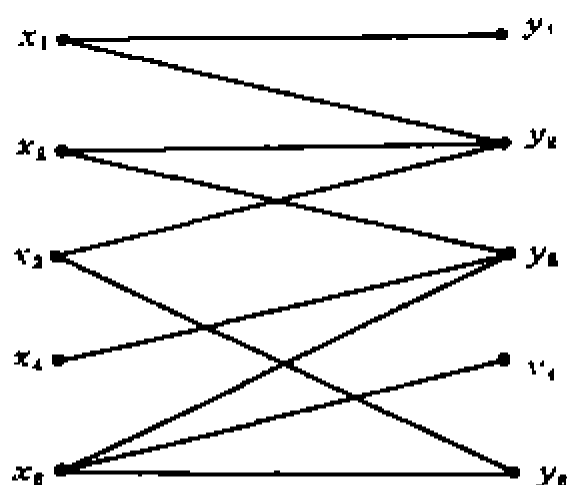


图 7-3-2

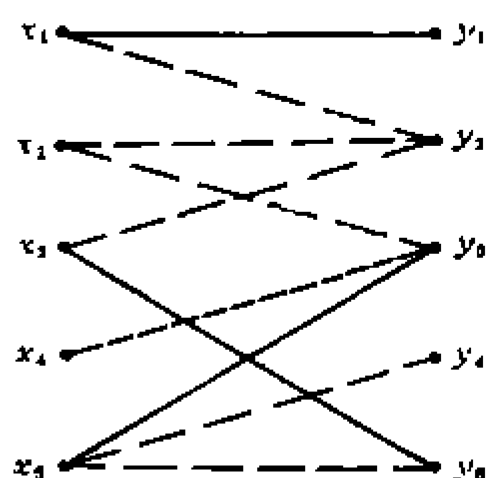


图 7-3-3

从而找到 y_2 为非饱和点和可增广道路(图 7-3-3 中箭头所示):

$$P=x_2y_3x_5y_5x_3y_2。$$

作 $M\leftarrow M\oplus E(p)$得新的匹配如图 7-3-4 所示。

第三步:X 未饱和,x_4 为非饱和点,从 x_4 点出发经过下列过程:

V_1: $\{x_4\}\rightarrow\{x_4,x_2\}\rightarrow\{x_1,x_2,x_3\}\rightarrow\{x_4,x_2,x_3,x_5\}$,

V_2: $\varnothing\rightarrow\{y_3\}\rightarrow\{y_3,y_2\}\rightarrow\{y_3,y_2,y_5\}\rightarrow\{y_3,y_2,y_5,y_4\}$。

从而得到非饱和点 y_4,以及从 x_4 到 y_4 的一条可增广道路:

$$P=x_4y_3x_2y_2x_3y_5x_5y_4。$$

作:$M\leftarrow M\oplus E(P)$得新的匹配 M',如图 7-3-5 所示。

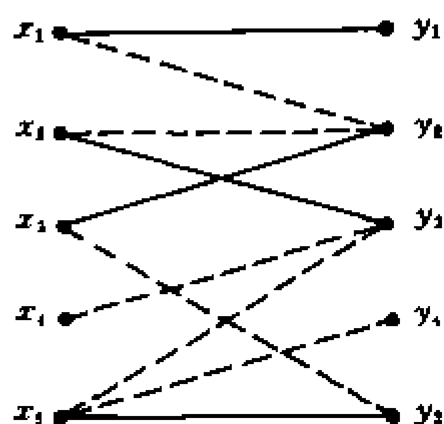


图 7-3-4

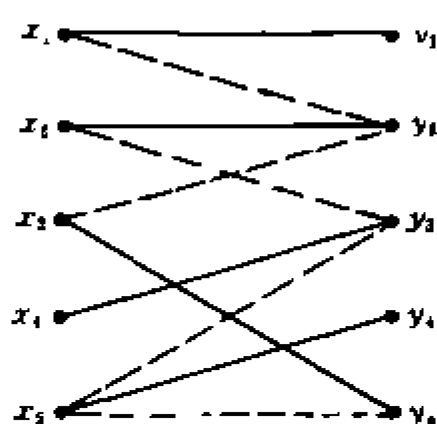


图 7-3-5

第四步：X 已全部饱和，故结束。

§4 最佳匹配

首先我们给出讨论最佳匹配问题用到的 Konig 定理。

设矩阵：

$$A = (a_{ij})_{n\times l}$$

的元素 a_{ij} 只取 0 或 1 两个值。

对矩阵 A 可以适当地选择若干行和列，使得这些行和列复盖了矩阵 A 的所有元素 1。办法也不是唯一的。其中有一行列数（指行数列数的和）最少的设为 m，显然有：

$$m \leqslant \min(n, l)。$$

另一方面，从矩阵 A 中可以找到一些元素 1，使得两两不在同一行，也不在同一列，取法也很不一样，其中元素 1 的数目的最大者设为 M，换句话说，从矩阵 A 中最多可以选取 M 个元素 1，这 M 个元素 1 分布在矩阵 A 中，使得每行每列最多只有其中的一个，此外的元素 1 必和这 M 个元素同行或同列。也就是说任意 $M+1$ 个 1 必有同行或同列的。

Konig 定理 对于上述的矩阵 A，恒有：

$$m = M。$$

证明 显然有 $m \geqslant M$。因为 m 条线既然复盖住了所有的元素 1，也必然复盖住这 M 个元素，故只要证 $M \geqslant m$，定理便得到证明。

设这 m 个行、列中有 r 行、c 列，即

$$m = r + c。$$

不妨假设这 r 行分别为第 $i_1, i_2, \cdots, i_r$ 行，c 列分别为第 $j_1, j_2, \cdots, j_c$ 列。

对应于其中第 i 行有下标的集合：

$$s_i \triangleq \{l \mid a_{il} = 1,\ l \neq j_1, j_2, \cdots, j_c\},$$

于是有 $s_{i_1}, s_{i_2}, \cdots, s_{i_r}$。从 $s_{i_1}, s_{i_2}, \cdots, s_{i_r}$ 中任取 $k(<r)$ 个，它所包含的不同的元素个数不少于 k。否则这 k 行可用少于 k 的列所取代，这与 $m=r+c$ 是最少行列数的假设相矛盾。

根据 Hall 定理，从这 r 行中可选取 r 个 1 元素，使得这 r 个 1 分别在不同的 r 列上，也就是利用匈牙利算法，使得这 r 行与 r 列相匹配。当然这 r 个 1 元素也不在第 $j_1, j_2, \cdots, j_c$

列中任何一列。类似的理由，可从第 $j_1,j_2,\cdots,j_c$ 列中选出 c 个 1 元素，分别在不同的 c 行上，应特别指出的是这 c 个元素 1 不属于第 $i_1,i_2,\cdots,i_r$ 行中的任何一行。故得 $r+c$ 个 1 两两不在同一行或同一列上。故 $M\geqslant m$。证毕。

前面我们讨论了 m 个人，n 项工作的任务安排问题，并未考虑成本和利润之类的问题。因为并非任何人做什么工作成效都是一样，所以实际安排中考虑成本和利润是很自然的。如果设第 i 个人从事第 j 项工作所得的利润为 c_{ij}，于是有利润矩阵：

$$C=(c_{ij})_{m\times n}。$$

其中 $c_{ij}\geqslant 0$。

不妨假定 $m=n$。实际上不同的安排方案，对应于 $1,2,\cdots,n$ 不同的排列 (j)—$(j_1j_2\cdots j_n)$。即将排列 $(j_1j_2\cdots j_n)$ 对应于第 i 个人从事第 j_i 项任务，$i=1,2,\cdots,n$。所谓最佳匹配，即求 $1,2,\cdots,n$ 的一个排列 $(j_1j_2\cdots j_n)$，使得

$$w(j)=\sum_{i=1}^{n}c_{ij_i}$$

取最大值。类似的定义，若矩阵 C 是成本矩阵，则求排列 (j) 使 $w(j)$ 取最小值。

若对点 x_i 给以标号 $l(x_i)$，y_j 点给以标号 $l(y_j)$，$l(x_i)$ 和 $l(y_j)$ 都是正数。使之满足条件：

$$l(x_i)+l(y_j)\geqslant c_{ij},\quad i,j=1,2,\cdots,n。$$

显然满足这条件的标号 l 也不是唯一的，是一个集合。于是有：

定理 设 $C=(c_{ij})_{n\times n}$ 的元素 c_{ij} 为正数，排列 $(j_1,j_2,\cdots,j_n)$ 使得

$$\sum_{i=1}^{n}c_{ij_i}$$

取最大值，则存在标号 $l(x_i)$ 和 $l(y_j)$，$i,j=1,2,\cdots,n$，使得

$$\max_{(j)}\sum_{i=1}^{n}c_{ij_i}=\min_{(l)}\sum_{i=1}^{n}[l(x_i)+l(y_i)],$$

而且满足：

$$l(x_i)+l(y_{j_i})=c_{ij_i}。$$

证明 令

$$m=\min_{(l)}\sum_{i=1}^{n}[l(x_i)+l(y_i)],$$

$$M=\max_{(j)}\sum_{i=1}^{n}c_{ij_i}。$$

取初始标号：

$$l(x_i)=\max_{j}(c_{ij}),$$

$$l(y_j)=0,\quad i,j=1,2,\cdots,n。$$

则

$$l(x_i)+l(y_j)\geqslant c_{ij}。$$

所以

$$\sum_{j=1}^{n}[l(x_i)+l(y_{j_i})]=\sum_{i=1}^{n}[l(x_i)+l(y_i)]$$
$$\geqslant\sum_{i=1}^{n}c_{ij_i}。$$

这个式子说明对于满足条件：

$$l(x_i)+l(y_j)\geqslant c_{ij}$$

的任何标号 l，则和 $\sum_{i=1}^{n}[l(x_i)+l(y_{p_i})]$ 不小于任何方案所得的利润 $\sum_{i=1}^{n}c_{ij_i}$，当然也不小于最大利润，所以

$$m\geqslant M。$$

作矩阵

$$B=(b_{ij})_{n\times n},$$

其中

$$b_{ij}=[l(x_i)+l(y_j)]-c_{ij},$$
$$i,j=1,2,\cdots,n。$$

如若 B 矩阵存在一组 n 个零元素，其中没有两个零元素在同一行或同一列中出现，则按这组 0 元素的位置 (i,j) 有：

$$\sum_{i=1}^{n}[l(x_i)+l(y_j)]=\sum_{i=1}^{n}c_{ij},$$

即

$$m=M。$$

由此便可得到问题的解。

如若不然，B 中只能找到 k 个 0 元素，使得其中任意两个零元素不在同一行或同一列上。根据 Konig 定理，一定可以找到总数为 k 的行和列，使得所有的零元素都在这总数为 k 的行和列上。假设这 k 个行和列分别是：

(1) 第 $i_1,i_2,\cdots,i_r$ 行，

(2) 第 $j_1,j_2,\cdots,j_c$ 列，$(r+c=k)$。

则对标号 l 作如下的修改：

(a) $x_{i_1},x_{i_2},\cdots,x_{i_r}$ 点的标号不动，其它 $n-r$ 个属于 X 的点标号减 1 作为该点的新标号；

(b) $y_{j_1},y_{j_2},\cdots,y_{j_c}$ 点的标号加 1 作为该点的标号，此外的 $n-c$ 个属于 Y 的点的标号不变。

这里 $X=(x_1,x_2,\cdots,x_n)$，$Y=(y_1,y_2,\cdots,y_n)$。

经过以上的修改后得一新的标号 l^*。

令：

$$X_r=\{x_{i_1},x_{i_2},\cdots,x_{i_r}\},$$
$$Y_c=\{y_{j_1},y_{j_2},\cdots,y_{j_c}\}。$$

则有：

(1) 当 $x_i\in X_r$，$y_j\in Y_r$ 时有：

$$l^*(x_i) + l^*(y_j) = l(x_i) + l(y_j) + 1 > c_{ij}。$$

(2) 当 $x_i \notin X_r$, $y_j \notin Y_c$ 时有:

$$l^*(x_i) + l^*(y_j) = l(x_i) + l(y_j) - 1 \geqslant c_{ij}。$$

(因为这时 $l(x_i)+l(y_j)>c_{ij}$)。

(3) 当 $x_i \in X_r$ 但 $y_j \notin Y_c$ 时(或 $x_i \notin X_r$ 但 $y_j \in Y_c$ 时)有:

$$l^*(x_i) + l^*(y_j) = l(x_i) + l(y_j) \geqslant c_{ij}。$$

故对于经过修改后的新标号 l^* 有:

$$\begin{aligned}&\sum_{i=1}^{n}[l^*(x_i) + l^*(y_i)] \\ &= \sum_{i=1}^{n}[l(x_i) + l(y_i)] - (n-r) + c \\ &= \sum_{i=1}^{n}[l(x_i) + l(y_i)] - (n-r-c)\end{aligned}$$

显然对于新标号 l^*,关系式 $l^*(x_i)+l^*(y_j)\geqslant c_{ij}$ 依然成立。

由于 $\qquad r+c<n$,

故 $\qquad n-(r+c)>0$,

即 $\qquad \sum_{i=1}^{n}[l^*(x_i) + l^*(y_i)]$

比 $\sum_{i=1}^{n}[l(x_i) + l(y_i)]$ 减少了 $n-r-c$。故当矩阵 $\boldsymbol{C}$ 的元素是整数时,只要经过有限次地利用上述步骤,便可达到 $\sum_{i=1}^{n}[l(x_i) + l(y_i)]$ 的最小值,从而给出最佳匹配。证毕。

事实上上面的证明是构造性的,即证明的过程也就是构造算法的过程。

上述步骤可以推广到矩阵 $\boldsymbol{C}$ 的元素为有理数的场合。

现举一例说明标号 l 的修改过程。设矩阵

$$\boldsymbol{C} = \begin{pmatrix} 3 & 5 & 5 & 4 & 1 \\ 2 & 2 & 0 & 2 & 2 \\ 2 & 4 & 4 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 3 & 3 \end{pmatrix}\begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{matrix}$$
$$\qquad\; y_1 \;\; y_2 \;\; y_3 \;\; y_4 \;\; y_5$$

是利润矩阵,x_i 为原料,y_j 为产品,$i,j=1,2,3,4,5$。c_{ij}为用 x_i 作 y_j 产品的原料所得的利润,求一最佳匹配使所得利润最大。

(1) 给初始标号如下:

$$l(y_1) = l(y_2) = l(y_3) = l(y_4) = l(y_5) = 0,$$
$$l(x_1) = 5,\ l(x_2) = 2,\ l(x_3) = 4,\ l(x_4) = 1,\ l(x_5) = 3。$$

作矩阵 $\boldsymbol{B}=(b_{ij})_{5\times5}$使得:

$$b_{ij} = l(x_i) + l(y_j) - c_{ij}。$$

得

$$
\begin{array}{c}
 \\ l(x)\\ \downarrow \\ 5\\2\\4\\1\\3\\ \\
\end{array}
\begin{array}{c}
 \\ \\ \\ \left[\begin{array}{ccccc}
2 & 0 & 0 & 1 & 4\\
0 & 0 & 2 & 0 & 0\\
2 & 0 & 0 & 3 & 4\\
1 & 0 & 0 & 1 & 1\\
2 & 1 & 2 & 0 & 0
\end{array}\right] \\
\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \leftarrow l(y)\end{array}
\end{array}
$$

(2) 从上可见矩阵 B 的所有零元素可以被第 2、5 行和第 2、3 列所覆盖(如箭头所示的虚线),故对标号 l 修改如下:

$$l(x_1)=5-1=4,\ l(x_3)=4-1=3,\ l(x_4)=1-1=0;$$

$$l(y_2)=0+1=1,\ l(y_3)=0+1=1。$$

对应于新的标号 l 的矩阵 B 如下:

$$
B=\begin{array}{c}4\\2\\3\\0\\3\\ \end{array}
\begin{array}{c}
\left[\begin{array}{ccccc}
1 & 0 & 0 & 0 & 3\\
0 & 1 & 3 & 0 & 0\\
1 & 0 & 0 & 2 & 3\\
0 & 0 & 0 & 0 & 0\\
2 & 2 & 3 & 0 & 0
\end{array}\right]\\
\begin{array}{ccccc} 0 & 1 & 1 & 0 & 0\end{array}
\end{array}
$$

现在已不能用少于数目 5 的行和列复盖住矩阵 B 的所有元素。

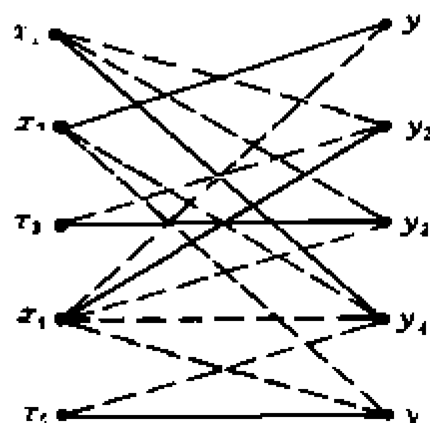


图 7-4-1

(3) 依 B 矩阵的零元素所在的行、列作图 7-4-1 所示的二分图。比如第一行有三个零元素分别在第 2、3、4 列上,则取 x_1y_2、x_1y_3、x_1y_4,其余类推。利用匈牙利算法找到匹配如图 7-4-1 中的实线所示,故

$$c_{14}+c_{21}+c_{33}+c_{42}+c_{55}=4+2+4+1+3=14$$

为最高利润。

§5 最佳匹配的算法及例

上节的定理给我们提供了求最佳匹配的算法,它的基本思想是按一定的办法修改标号 l,使得和:

$$\sum_{i=1}^{n}[l(x_i)+l(y_i)]$$

不断下降,直到给出问题的解为止。标号 l 的唯一要求是:

$$l(x_i)+l(y_j)\geqslant c_{ij},\ i,j=1,2,\cdots,n。$$

有时候最佳匹配则要求找到排列$(j_1j_2\cdots j_n)$使得:

$$w = \sum_{i=1}^{n} c_{ij_i}$$

取极小值，讨论的办法类似，这里不重复，留给读者思考。现给出另一算法如下。

图 7.5.1 是算法的流程图。从中可以看出与匈牙利算法颇有相似之处。

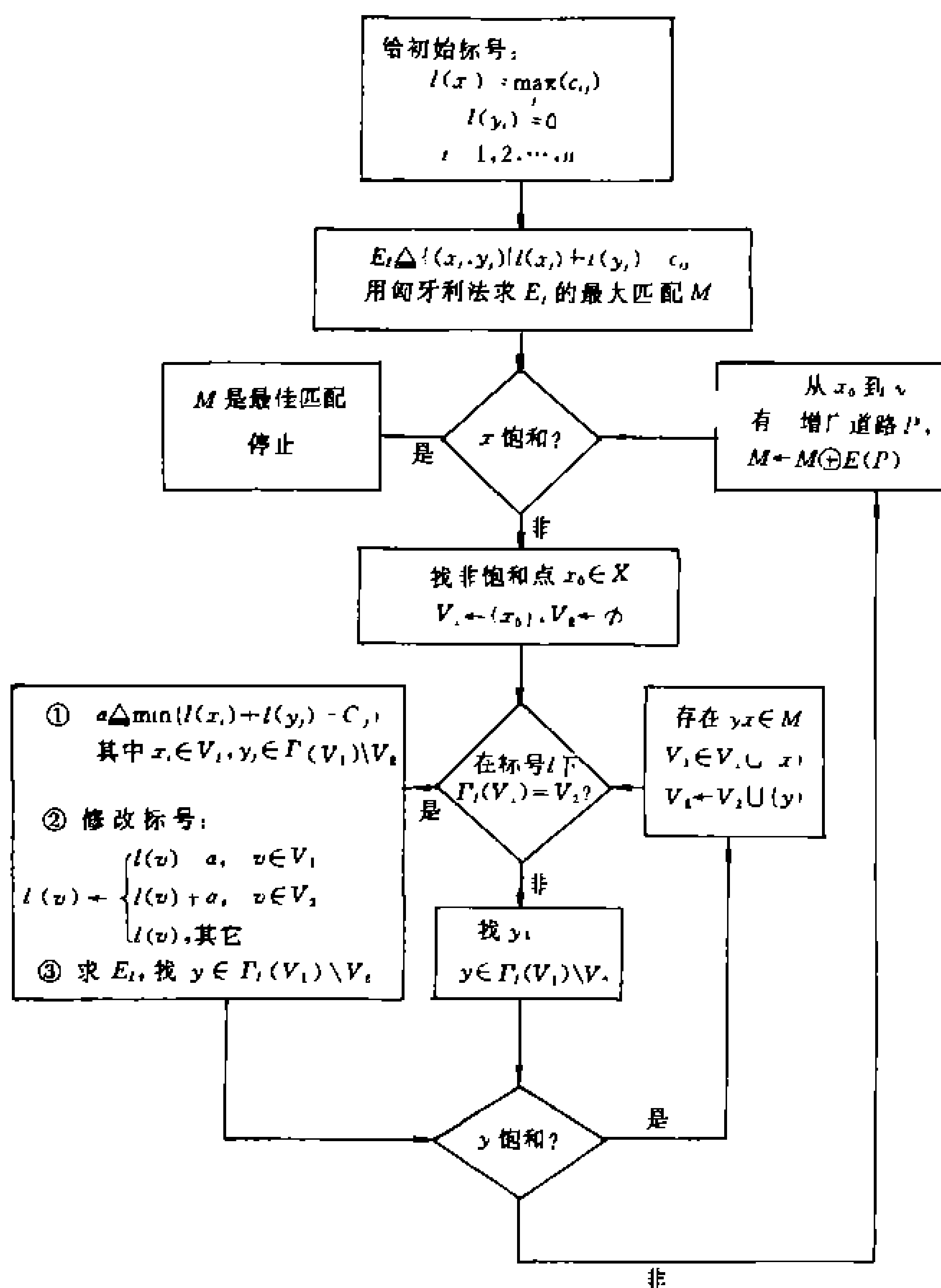


图 7.5.1

(1) 给初始标号：

$$l(x_i) \leftarrow \max_j\{c_{ij}\}, l(y_i) \leftarrow 0, i = 1,2,\cdots,n。$$

(2) 求 $E_l \triangleq \{(x_i,y_j) \mid l(x_i)+l(y_j)=c_{ij}\}$，并用匈牙利算法求 E_l 的最大匹配 M。

(3) 若 X 饱和则结束，否则在 X 中任找一非饱和点 x_0，$V_1 \leftarrow \{x_0\}$，$V_2 \leftarrow \varnothing$。

(4) 若在标号 l 下集合 E_l 中与 V_1 的点邻接的点集 $\Gamma_l(V_1)=V_2$，则转(5)，否则从 $\Gamma_l(V_1)\backslash V_2$ 中任找一点 y，转(6)。

(5) 计算

(a) $a=\min\{l(x_i)+l(y_j)-c_{ij}\}$，$x_i \in V_1$，$y_j \in \Gamma(V_1)\backslash V_2$。

(b) 对标号 l 作如下修改

$$
l(v) \leftarrow \begin{cases} l(v)-a, & v \in V_1 \\ l(v)+a, & v \in V_2 \\ l(v), & \text{其它。} \end{cases}
$$

(c) 求 E_l，在 $\Gamma_l(V_1)\backslash V_2$ 中任找一点 y。

(6) 若 y 饱和转(7)，否则作

【从 x_0 到 y 找一条可增广道路 P，作 $M \leftarrow M \oplus E(P)$，转(3)。】

(7) 因 y 饱和，故存在 $(y,x) \in M$，

作【$V_1 \leftarrow V_1 \cup \{x\}$，$V_2 \leftarrow V_2 \cup \{y\}$，转(4)。】

例：已知利润矩阵

$$
C=\begin{pmatrix} 5 & 7 & 7 & 6 & 3 \\ 4 & 4 & 0 & 4 & 4 \\ 4 & 6 & 6 & 3 & 0 \\ 0 & 3 & 3 & 0 & 0 \\ 3 & 4 & 3 & 5 & 5 \end{pmatrix},
$$

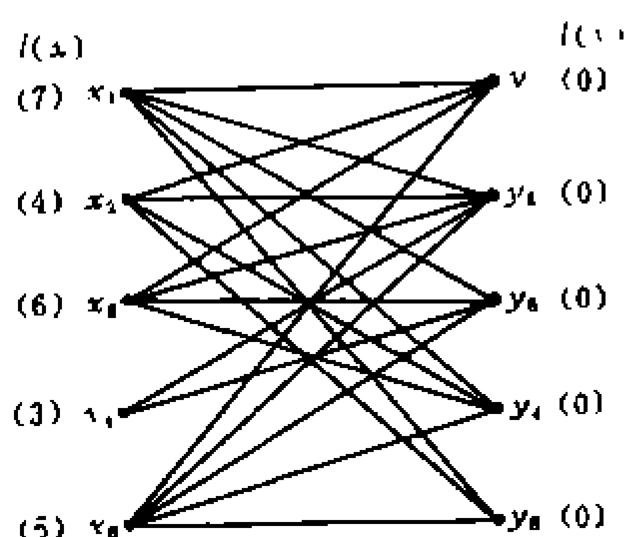


图 7-5-2

求最佳匹配。

(1) 给初始标号：

$l(x_1)=\max\limits_j c_{1j}=\max(5,7,7,6,3)=7$，

$l(x_2)=\max\limits_j c_{2j}=\max(4,4,0,4,4)=4$，

$l(x_3)=\max\limits_j c_{3j}=\max(4,6,6,3,0)=6$，

$l(x_4)=\max\limits_j c_{4j}=\max(0,3,3,0,0)=3$，

$l(x_5)=\max\limits_j c_{5j}=\max(3,4,3,5,5)=5$，

$l(y_1)=l(y_2)=l(y_3)=l(y_4)=l(y_5)=0$

(2) 在标号 l 下得

$E_l=\{x_1y_2, x_1y_3, x_2y_1, x_2y_2, x_2y_4, x_2y_5, x_3y_2, x_3y_3, x_4y_2, x_4y_3, x_5y_4, x_5y_5\}$。

不难发现 E_l 与上节中的矩阵 $\boldsymbol{B}$ 存在关系，这时

$$
\boldsymbol{B}=\begin{pmatrix} 2 & 0 & 0 & 1 & 4 \\ 0 & 0 & 4 & 0 & 0 \\ 2 & 0 & 0 & 3 & 6 \\ 3 & 0 & 0 & 3 & 3 \\ 2 & 1 & 2 & 0 & 0 \end{pmatrix},
$$

而 $b_{12}, b_{13}, b_{21}, b_{22}, b_{24}, b_{25}, b_{32}, b_{33}, b_{42}, b_{43}, b_{54}, b_{55}$ 恰好为零元素。

(3) 用匈牙利法求图 7-5-3 的最大匹配，匹配的边用实线给出。

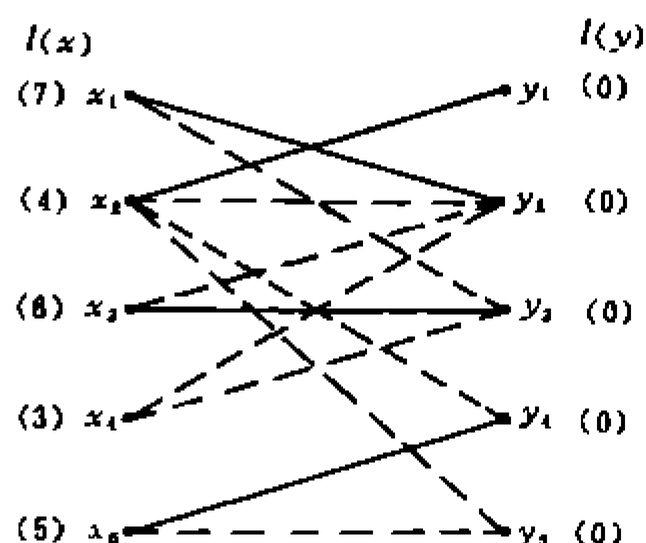


图 7-5-3

(4) x_4 未饱和，$V_1 \leftarrow \{x_4\}, V_2 \leftarrow \varnothing$。

$$\Gamma_l(V_1) \neq V_2,\ y_3 \in \Gamma_l(V_1)\backslash V_2,\ (y_3, x_3) \in M。$$

所以 $V_1 = \{x_4, x_3\},\ V_2 = \{y_3\}$，

$\Gamma_l(V_1) \neq V_2,\ y_2 \in P_l(V_1)\backslash V_2, (y_2, x_1) \in M$。

所以 $V_1 - \{x_4, x_3, x_1\},\quad V_2 - \{y_3, y_2\}$。

$\Gamma_l(V_1) - V_2$。

(5) $\Gamma(V_1)\backslash V_2 - \{y_1, y_4, y_5\}$，

$$a - \min_{\substack{x_i \in V_1 \\ y_j \in P(V_1)\backslash V_2}} \{l(x_i) + l(y_j)\quad c_{ij}\}$$

$$= \min_{\substack{i=1,3,4 \\ j=,,4,5}} \{l(x_i) + l(y_j) - c_{ij}\} - 1。$$

(6) 重新标号：

$l(x_1) = 7 - 1 - 6,\ l(x_3) = 6 - 1 = 5,\ l(x_4)\quad 1 - 1 - 0,$

$l(y_2) - 0 + 1 - 1,\ l(y_3) - 0 + 1 = 1$。

(7) 在新的标号 l 下，E_l 增加了一条边 x_1y_4，减少一条边 x_2y_2，如图 7-5-4 所示。

(8) 在新的标号 l 下，存在 $y_4 \in \Gamma_l(V_1)\backslash V_2$，$y_4$ 已饱和，$V_1 - \{x_4, x_3, x_1, x_5\}, V_2 - \{y_3, y_2, y_4\}$。由于 $\Gamma_l(V_1) \neq V_2$，存在 $y_5 \in \Gamma_l(V_1)\backslash V_2$，而且 y_5 非饱和，故存在一条可增广道路 P：

$$P = x_4y_3x_3y_2x_1y_4x_5y_5。$$

作 $M \leftarrow M \oplus E(P)$ 得图 7 5 5。

(9) X 已饱和，$x_1y_4,\ x_2y_1,\ x_3y_2,\ x_4y_3,\ x_5y_5$ 是最佳匹配。

$$w - c_{14} + c_{21} + c_{32} + c_{43} + c_{55} = 6 + 4 + 6 + 3 + 5 - 24。$$

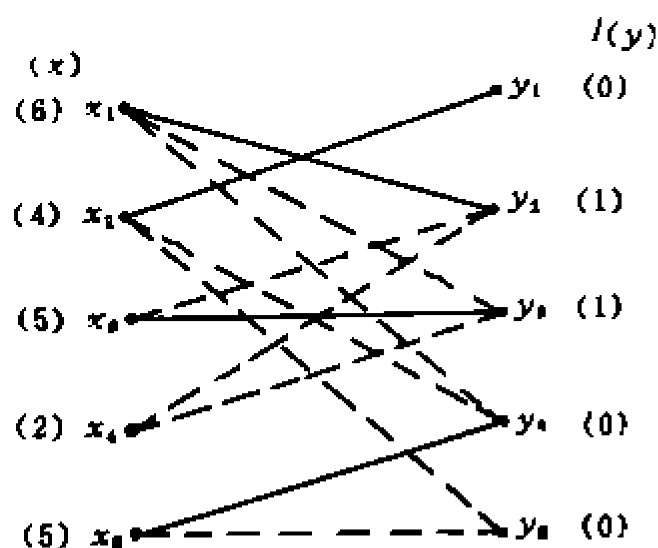


图 7-5-4

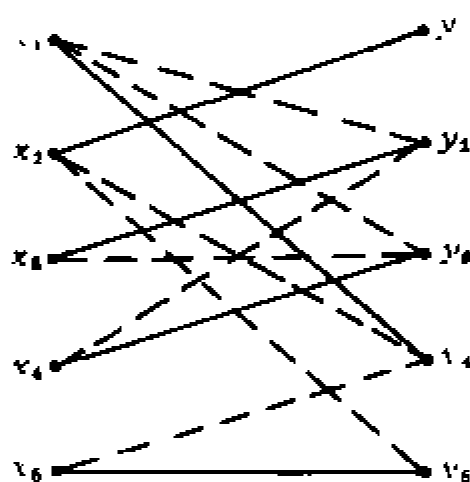


图 7-5-5

§6 色数问题

前面我们曾经提到过图的着色问题，不过只是就平面图讨论了四色、五色问题。下面将研究一般的着色问题，分为图的顶点着色和边着色两类。已知图 $G=(V,E)$，其中 $|V|=n$，$|E|=m$，对图 G 的所有顶点进行着色时，要求相邻的两顶点的颜色不一样，问最少要几种颜色？这就是所谓的顶点着色问题。若对图 G 的所有的边进行着色，要求与同一顶点相关联的边有不同的颜色，问最少要多少种颜色？这就是所谓的边的着色问题。这些问题的提出是有其实际背景的。

（1）课程考试安排问题：设某一学校有 n 种课程由学生选修，学期终了要进行考试，当然每个学生每场只能参加一门课程的考试。试问这次考试最少要进行几场？显然同一个学生修的两门课的考试不可以在同一时间进行。当然没有相同的学生的不同课程是可以在同一时间进行考试了。

在平面上取 n 个顶点 $v_1,v_2,\cdots,v_n$ 分别表示这 n 门课程，若有同学同时选了课程 i 和课程 j，则过 v_i,v_j 点连一条边，结果得一有 n 个顶点的图 G。考试的安排问题相当于图 G 的顶点着色问题。着同一颜色的顶点对应的课程可同时进行考试。使图 G 的相邻顶点有不同颜色的最少颜色数目，便是进行考试的最少场数。

（2）物资储存问题：设有 n 种物资要存放在仓库里，但有的物资不能放在同一房间里，否则将引起损坏，甚至发生危险。问存放这 n 种物资最少要几个房间？

和问题(1)相类似，在平面上取 n 个顶点 $v_1,v_2,\cdots,v_n$ 分别表示 n 种物资。设 v_i,v_j 为不能放在一起的两种物资，则过 v_i，v_j 点连一条边，得一 n 个顶点的图 G，于是问题又变为图 G 的顶点着色问题。

（3）时间表问题：设 $x_1,x_2,\cdots,x_m$ 为 m 个工作人员，$y_1,y_2,\cdots,y_n$ 为 n 种设备。设工作人员 x_i 对设备 y_j 提出要求，使用时间假定均以单位时间计算。自然每一个工作人员在同一个时间只能使用一种设备，某一种设备在同一时间里只能为一个工作人员所使用。问应如何合理安排，使得在尽可能短时间里满足工作人员的要求？

和前面讲的匹配问题类似，问题转为讨论关于 $X=\{x_1,x_2,\cdots,x_m\}$，$Y=\{y_1,y_2,\cdots,y_n\}$ 的二分图 G。工作人员 x_i 要求使用设备 y_j，每单位时间对应一条从 x_i 到 y_j 的边，这样所得的二分图 G 过 x_i,y_j 的边可能不只一条。问题变为对所得的二分图 G 的边着色问题。有相同颜色的边可以安排在同一时间里。

前面提出了点的着色与边的着色两类问题，然而边的着色问题可以转换为相应的点的着色问题。例如要对图 7-6-1 的诸边（实线的边）进行着色，要求与同一顶点关联的边有不同的颜色。

对应于图 7-6-1 的图，作一图使满足下列条件：

(1) 图 7-6-1 的每一条边（设为 e_i）对应一顶点（设为 v_i）；

(2) 当图 7-6-1 的两条边 e_i,e_j 有一共同顶点时，则过 v_i,v_j 点引一条边，结果得一图 7-6-2 即图 7-6-1 的虚线图。

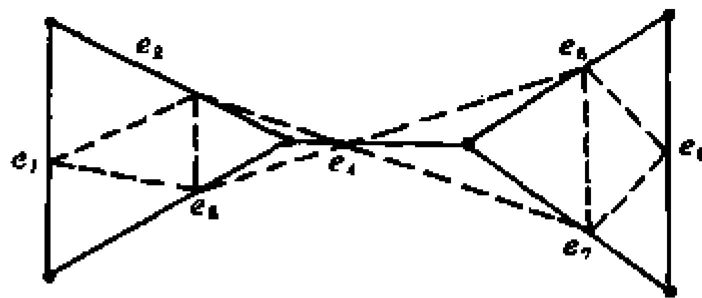


图 7-6-1

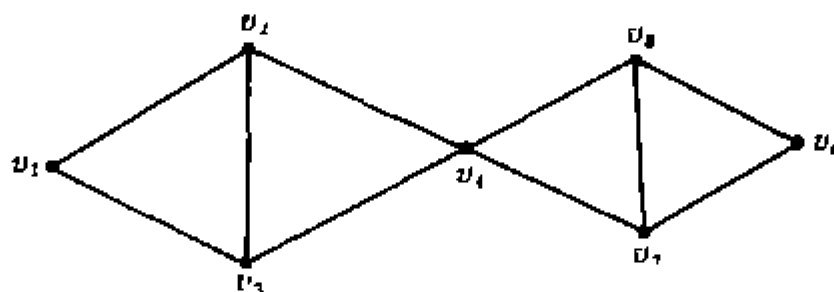


图 7-6-2

下面我们来讨论色数及其性质：

对图 $G=(V,E)$ 的所有顶点进行着色，后面若不特别说明，都是指使相邻的两顶点有不同的颜色。对图 $G=(V,E)$ 的顶点 V 进行着色所需最少的颜色数目用 $\chi(G)$ 表示，称为是图 G 的色数。显然有

(1) 图 G 只有孤立点时，$\chi(G)=1$；

(2) n 个顶点的完全图 G 有 $\chi(G)=n$；

(3) 若图 G 是 n 个顶点的回路时，则

$$\chi(G)=\begin{cases}2, & n\ 为偶数；\\ 3, & n\ 为奇数。\end{cases}$$

(4) 图 G 是顶点数超过 1 的树时，$\chi(G)=2$。

(5) 若图 G 是二分图，则 $\chi(G)=2$。

定理 图$G=(V,E)$的色数$\chi(G)=2$的充要条件是：(a) $|E|>1$；(b) G不存在边数为奇数的回路。

证明 若G是连通图，且不存在边数为奇数的回路。任从图G中找一棵树T，显然作为G的子图T有$\chi(T)=2$。

在树T上每加一条余树边时，便获得图G的一条回路。根据图G不存在边数为奇数的回路，故所有余树边的两端点颜色不同。这就证明了若图G设有奇数边的回路时，$\chi(G)=2$。反之，若图G有奇数边的回路时，$\chi(G)\geqslant 3$。证毕。

定理 若图$G=(V,E)$，$d=\max\limits_{v_i\in V}\{d(v_i)\}$，

则
$$\chi(G)\leqslant d+1。$$

证法1 只要证对于这样的图，$d+1$种颜色足够了。任取一顶点，着以$d+1$种颜色中某一颜色，再取另一个未着色的顶点，着以与相邻已着色的顶点不同的颜色。由于邻近的顶点最多只有d个，故这个颜色总是可以找到的。继续这个步骤直到所有的顶点都着上颜色为止。定理证毕。

证法2 这个定理还可以利用归纳法来证明。设$n\triangleq|V|$。显然$n=1$时，图G只有一顶点，此时$d=0$，而$\chi(G)=1$，定理成立。假设定理对$|V|=n-1$时为真。若从图G中去掉一顶点v，及以v点为顶点的所有边，得一子图$G_1=(V_1,E_1)$。根据假设$\chi(G_1)\leqslant d_1+1$，其中$d_1=\max\limits_{v_i\in V(G_1)}d(v_i)$。但$d_1\leqslant d$，故有$\chi(G_1)\leqslant d+1$。设用$d+1$种颜色对子图$G_1$着色。若$Adj(v)=\{v_{k_1},v_{k_2},\cdots,v_{k_l}\}$，设$c_1,c_2,\cdots,c_l$分别是$v_{k_1},v_{k_2},\cdots,v_{k_l}$点所着的颜色。因$l\leqslant d$故从$d+1$种颜色中必然可以找到$c_1,c_2,\cdots,c_l$以外的颜色，设为$c_{l+1}$，给$v$着上$c_{l+1}$即可。定理证毕。

下面给出图的顶点着色的算法。假定图G的顶点为v_1，$v_2,\cdots,v_n$，用来染色的颜色为$c_1,c_2,\cdots,c_n$。

(1) $i=1,2,\cdots,n$作$\dot{C}_i=\{c_1,c_2,\cdots,c_i\}$，$j\leftarrow 1$。

(2) 若c_l是$\dot{C}_j$的第一个颜色，给v_j着c_l色。

(3) $\forall h>j$，若v_j和v_h相邻则作

【$\dot{C}_h\leftarrow\dot{C}_h\setminus\{c_l\}$，$j\leftarrow j+1$，转(4)。】，

(4) 若$j\leftarrow n$则转(2)，否则转(5)。

(5) 输出各顶点的颜色，停止。

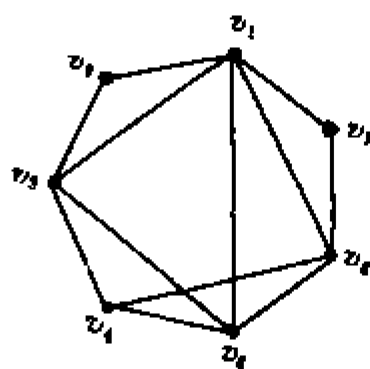


图 7-6-3

看一个例子，见图7-6-3(a)。

这里必须说明一下，下面的(1)，(2)，(3)，表示执行算法中的步骤(1)，(2)，(3)。

(1) $\dot{C}_1=\{c_1\}$，$\dot{C}_2=\{c_1,c_2\}$，$\cdots$，$\dot{C}_7=\{c_1,c_2,\cdots,c_7\}$。$j\leftarrow 1$。

(2) v_1点着以c_1。

(3) $\dot{C}_2=\{c_2\}$，$\dot{C}_3=\{c_2,c_3\}$，$\dot{C}_4=\{c_1,c_2,c_3,c_4\}$，$\dot{C}_5=\{c_2,c_3,c_4,c_5\}$，

$\dot{C}_6=\{c_2,c_3,c_4,c_5,c_6\}$，$\dot{C}_7=\{c_2,c_3,c_4,c_5,c_6,c_7\}$，$j=2$。

(2) v_2着以c_2。

(3) 由于 v_3 与 v_2 相邻，故 $\dot{C}_3=\{c_3\}$。$j=3$。

(2) v_3 着以 c_3。

(3) v_4，v_5 与 v_3 相邻，故 $\dot{C}_4=\{c_1,c_2,c_4\}$，$\dot{C}_5=\{c_2,c_4,c_5\}$。$j=4$。

(2) v_4 着以 c_1。

(3) v_5，v_6 和 v_4 相邻，故

$\dot{C}_5=\{c_2,c_4,c_5\}$，$\dot{C}_6=\{c_2,c_3,c_4,c_5,c_6\}$不变。$j=5$。

(2) v_5 着以 c_2。

(3) v_6 与 v_5 相邻，故 $\dot{C}_6=\{c_3,c_4,c_5,c_6\}$。$j=6$。

(2) v_6 着以 c_3。

(3) v_7 和 v_6 相邻，故 $\dot{C}_7=\{c_2,c_4,c_5,c_6,c_7\}$。$j=7$。

(2) v_7 着以 c_2，$j=8$。

(3) v_1 和 v_4 着 c_1；v_2，v_5，v_7 着 c_2；v_3，v_6 着 c_3。

本算法并不能保证给出的着色方案用的颜色是最少的。二分图的顶点着色只要两种颜色就够了，但用本算法就可能超过。举例如下。

下面介绍一种最大度数优先的 Welsh-Powell 算法。要求顶点序列 $v_1,v_2,\cdots,v_n$ 满足

$$d(v_1)\geqslant d(v_2)\geqslant\cdots\geqslant d(v_n)。$$

其余的仍照算法的步骤执行。根据规定必须对顶点的顺序重新定义。

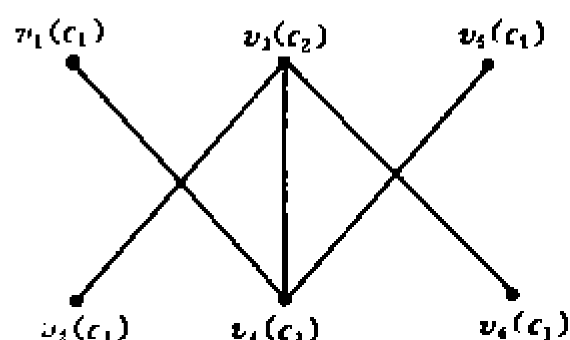


图 7-6-4

$v_1'=v_3$，$v_2'=v_4$，$v_3'=v_1$，$v_4'=v_5$，

$v_5'=v_2$，$v_6'=v_6$。

对 v_1'，v_2'，$\cdots$，v_6' 执行

(1) $\dot{C}_1=\{c_1\}$，$\dot{C}_2=\{c_1,c_2\}$，$\cdots$，$\dot{C}_6=\{c_1,c_2,\cdots,c_6\}$，$j=1$。

(2) v' 着以 c_1，即 v_3 着 c_1 色。

(3) $\dot{C}_2=\{c_2\}$，$\dot{C}_5=\{c_2,c_3,c_4,c_5\}$，$\dot{C}_6=\{c_2,c_3,\cdots,c_6\}$，$j=2$。

(2) v_3(即 v_4)点着 c_2。

(3) $\dot{C}_3=\{c_1,c_3\}$，$\dot{C}_4=\{c_1,c_3,c_4\}$，$j=3$。

(2) v_5'(即 v_1)点着以 c_1 色。

(3) $\dot{C}_4$，$\dot{C}_5$，$\dot{C}_6$ 保持不变，$j=4$。

(2) v_4'(即 v_5)点着 c_1 色。

(3) $\dot{C}_5$，$\dot{C}_6$ 不变，$j=5$。

(3) v_5'(即 v_2)点着以 c_2 色。

(3) $\dot{C}_6$ 不变，$j=6$。

(2) v_6'(即 v_6)点着以 c_2 色，$j=7$。

(6) v_1'(即 v_3)点着 c_1，

v_2'(即 v_4)点着 c_2，

v_3'(即 v_1)点着 c_3，

v_4'(即 v_5)点着 c_3，

v_5'(即 v_2)点着 c_2，

v_6'(即 v_6)点着 c_2。

还可以采用最小度最后的算法，即取 v_n 为顶点中度最小者，取 v_{n-1} 为子图 $G\setminus\{v_n\}$ 中度数最小者。依此类推得顶点序列

$$v_1, v_2, \cdots, v_n。$$

然后执行算法。

§7 独立集概念及其应用

对图 $G=(V,E)$ 的顶点进行着色的结果，把顶点 V 划分成若干不相交的子集，而且每一子集中的任意两点都不相邻，这样的顶点集合(指集合中任意两顶点都不相邻)叫做独立集。

设图 G 的某一顶点集合是独立集，但是任意增加一顶点就破坏它的独立性，则称这独立集为极大的独立集。

极大独立集不是唯一的，而且连顶点个数都不尽相同。以图 7-7-1 为例，$\{v_2, v_8\}$ 是图 7-7-1 的一个极大独立集，然而 $\{v_2, v_4, v_5, v_7\}$ 也是一个极大的独立集。

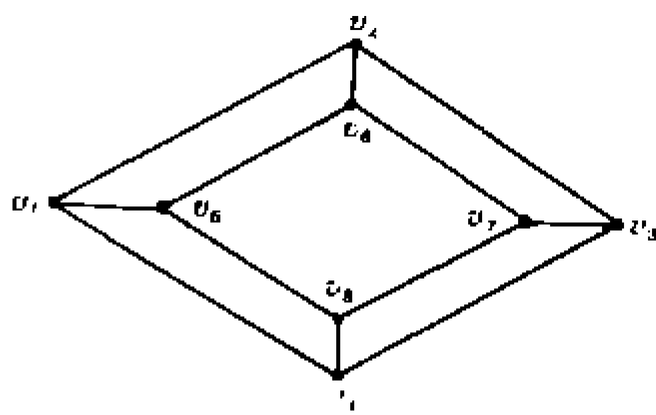


图 7-7-1

顶点数目最多的极大独立集的顶点数，表示为 $\beta(G)$。如图 7-7-1 的 $\beta(G)=4$。

在介绍独立集的应用之前，先介绍一下图的乘积的概念：

设已知图 $G^{(1)}=(V^{(1)}, E^{(1)})$，$G^{(2)}=(V^{(2)}, E^{(2)})$，其中

$$V^{(1)}=\{v_1^{(1)}, v_2^{(1)}, \cdots, v_{n_1}^{(1)}\},$$

$$V^{(2)}=\{v_1^{(2)}, v_2^{(2)}, \cdots, v_{n_2}^{(2)}\}。$$

图 $G=G^{(1)}\times G^{(2)}=(V,E)$ 称为图 G_1 与 G_2 的乘积，若满足：

(a) $V(G)=\{(v_i^{(1)}, v_j^{(2)}) \mid v_i^{(1)}\in V^{(1)}, v_j^{(2)}\in V^{(2)}\}$，

(b) $Adj((v_i^{(1)}, v_j^{(2)}))=\{(v_k^{(1)}, v_j^{(2)}) \mid v_k^{(1)}\in Adj(v_i^{(1)})\}$
$\quad\quad \vee \{(v_i^{(1)}, v_l^{(2)}) \mid v_l^{(2)}\in Adj(v_j^{(2)})\}\vee$
$\quad\quad \{(v_k^{(1)}, v_l^{(2)}) \mid v_k^{(1)}\in Adj(v_i^{(1)}), v_l^{(2)}\in Adj(v_j^{(2)})\}$。

式中 $Adj(v_i)$ 表示与 v_i 点相邻的点的集合。$(v_i^{(1)}, v_j^{(2)})$ 不是一条边，而是 $G^{(1)}\times G^{(2)}$ 图的

个顶点。

条件(a)说明图 $G=G_1\times G_2$ 的顶点是由 $V^{(1)},V^{(2)}$ 的直积。条件(b)说明图 G 的边包含如下几种情形：

(1) 当 G_1 图中 $v_k^{(1)}\in Adj(v_i^{(1)})$ 时，在 G 图中 $(v_k^{(1)},v^{(2)})\in Adj((v_i^{(1)},v_j^{(2)}))$。即在 G 图中 $v_i^{(1)}$ 与 $v_k^{(1)}$ 有边相连时，则 G 图顶点 $(v_i^{(1)},v_j^{(2)})$ 与 $(v_k^{(1)},v_j^{(2)})$ 有边相连。

(2) 在 G_2 图中若 $v_l^{(2)}\in Adj(v_j^{(2)})$，则

$$(v_i^{(1)},v_l^{(2)})\in Adj((v_i^{(1)},v_j^{(2)}))。$$

(3) 若在 G 图中 $v_k^{(1)}\in Adj(v_i^{(1)})$，在 G_2 图中 $v_l^{(2)}\in Adj(v_j^{(2)})$，则

$$(v_k^{(1)},v_l^{(2)})\in Adj((v_i^{(1)},v_j^{(2)}))。$$

例：

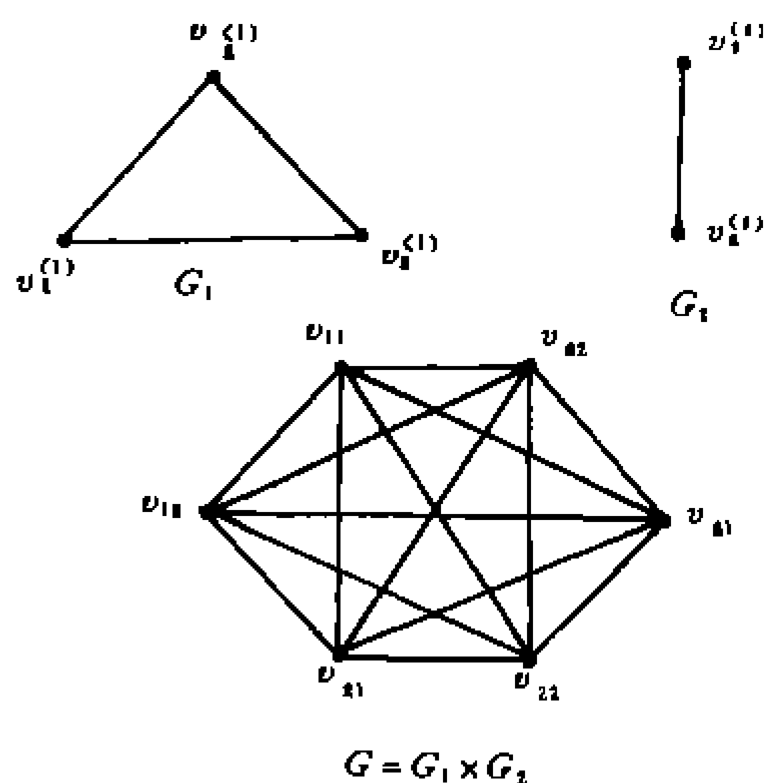


图 7-7-2

现在我们来看一个独立集应用的例子：

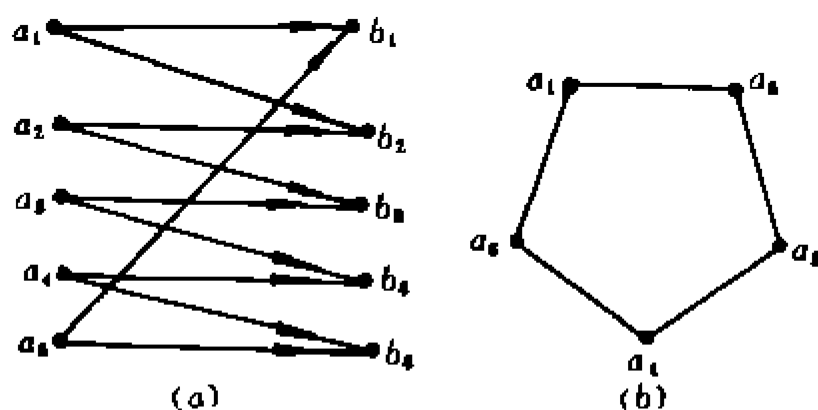


图 7-7-3

图 7-7-3表示某一通讯通道里，输入端有五个符号允许输入，设为 a_1, a_2, a_3, a_4, a_5，输出端可以获得 b_1, b_2, b_3, b_4, b_5 五种讯息。由于有干扰，若输入端为 a_1 时，输出端可能为 b_1，也可能为 b_2。a_2 同样也可能被接受为 b_2 或 c_2，等等。这样从输出端输出的讯息不能正确地断定输入端的讯息。为了正确无误地从输出端得到输入端的讯息，输入端只能选取 a_1, a_2, a_3, a_4, a_5 中的若干符号，而不能是它们的全体。

显然 $\beta(G)=2$。

图 7-7-3(b)说明 a_1 可能和 a_2 发生错乱，也可能与 a_5 发生错乱，a_2 可能和 a_1、也可能和 a_3 发生错乱，等等。所以为了使输出端输出的讯息正确无误地反映输入端的讯息，问题导致求图 7-7-3(b)的极大独立集。例如选取$\{a_1, a_3\}$，$\{a_1, a_4\}$，$\{a_2, a_5\}$等。

如果要传输两个以上的信息时，就得采用由几个符号组成的码。

从下表可以看出采用 $a_1a_1, a_2a_3, a_3a_5, a_4a_2, a_5a_4$ 这五个字符串可明确无误地表达 5 种讯息，即输出端讯息没有混淆不清的。可能有 20 个输出各不相同。例如 b_1b_2 只出现在第一行，别的地方再无出现。

输入端讯息	输出端讯息
a_1a_1	b_1b　b_1b_2　b_2b　b_2b_2
a_2a_3	b_2b_3　b_2b_4　b_3b_3　b_3b_4
a_3a_5	b_3b_5　b_3b_1　b_4b_5　b_4b_1
a_4a_2	b_4b_2　b_4b_3　b_5b_2　b_5b_3
a_6a_4	b_5b_4　b_5b_5　b_1b_4　b_1b_5

为了找出两个字符组成的使输出端给出明确无误的讯息问题，导致求图 $G\times G$ 的极大独立集。类似的理由，如果必要时可以求 $G\times G\times G$ 的极大独立集，等等。

前面介绍了极大独立集的概念，从定义可以看出：找一个极大独立集并不困难，从任意一点开始，然后每次增加一点，保证所加的点与已有的点均不相邻，这步骤反复进行，直到最后不能增加为止便得一极大独立集。下面介绍一种算法——通过布尔变量的运算找出所有的极大独立集。

若将图 $G=(V,E)$的每一个顶点 v_i 当作一个布尔变量，于是 $v_i\wedge v_j$ 表示包含 v_i 和 v_j 两点，这样图 G 的过 v_i, v_j 点的边对应一布尔积 $v_i\wedge v_j$。$v_i\vee v_j$ 表示以下三种情形：

(a) 或是包含一顶点 v_i；

(b) 或包含一顶点 v_j；

(c) 或包含 v_i, v_j 两点。根据以上约定，则有：

(1) $v_i\vee v_i=v_i$；

(2) $v_i\wedge v_i=v_i$；

(3) $v_i\vee(v_i\wedge v_j)=v_i$；

(4) $v_i\wedge(v_i\vee v_j)=v_i$。

(1)、(2)是显然的，(3)、(4)从下面的真值表可得而知。

v_i	v_j	$v_i \wedge v_j$	$v_i \vee (v_i \wedge v_j)$	$v_i \wedge (v_i \vee v_j)$
T	T	T	T	T
F	F	F	F	F
T	F	F	T	T
F	T	F	F	F

作布尔表达式：

$$\varphi = \bigvee_{(v_i, v_j) \in E} (v_i \wedge v_j)。$$

即 φ 中每一项 $v_i \wedge v_j$ 对应于 G 中的一条边，$\vee$ 是对所有的边求逻辑和。由 Morgan 法则：

(1) $\overline{(v_i \wedge v_j)} = \bar{v}_i \vee \bar{v}_j$，

(2) $\overline{(v_i \vee v_j)} = \bar{v}_i \wedge \bar{v}_j$。

从而可得

$$\bar{\varphi} = \bigwedge_{(v_i, v_j) \in E} (\bar{v}_i \vee \bar{v}_j)。$$

设

$$\bar{\varphi} = \varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_k$$

φ 与 $\bar{\varphi}$ 都是含有布尔变量 $v_1, v_2, \cdots, v_n$ 的表达式，因极大独立集不包含任何一条边的两端点，故表达式 φ 在任一极大独立集上取布尔值 0。反之，使 φ 取值 0 的点是独立集。即布尔表达式 φ 取布尔值 0 是独立集的充要条件。换句话说，使 $\bar{\varphi}$ 取布尔值 1 是独立集的充要条件。由于 $\bar{\varphi} = \varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_k$，只要其中任意一项为 1，设 $\varphi_1 = 1$，则 $\bar{\varphi} = 1$。故分别使 $\varphi_1, \varphi_2, \cdots, \varphi_k$ 取布尔值 1 的点都是极大独立集。

例：如图 7 7 4

$$\varphi = (v_1 \wedge v_2) \vee (v_1 \wedge v_3) \vee (v_1 \wedge v_4)$$
$$\vee (v_2 \wedge v_4) \vee (v_3, v_4) \vee (v_4 \wedge v_5)$$
$$\vee (v_4 \wedge v_6) \vee (v_5 \wedge v_6)。$$

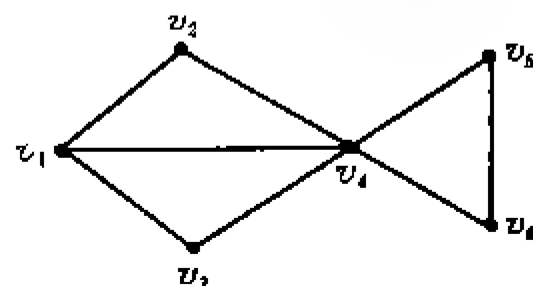


图 7-7-4

根据 Morgan 法则则有：

$$\bar{\varphi} = \overline{(v_{\ } \wedge v_2)} \wedge \overline{(v_{\ } \wedge v_1)} \wedge \overline{(v_1 \wedge v_4)} \wedge \overline{(v_2 \wedge v_4)}$$
$$\wedge \overline{(v_3 \wedge v_4)} \wedge \overline{(v_4 \wedge v_5)} \wedge \overline{(v_4 \wedge v_n)} \wedge \overline{(v_5 \wedge v_6)}。$$
$$= (\bar{v}_1 \vee \bar{v}_2) \wedge (\bar{v}_1 \vee \bar{v}_3) \wedge (v_1 \vee v_4) \wedge (v_2 \vee v_4)$$
$$- \wedge (v_3 \vee v_4) \wedge (v_4 \vee v_5) \wedge (v_4 \vee v_6) \wedge (\bar{v}_5 \vee \bar{v}_6)。$$

为了不致引起混淆而且方便起见，$v_i \wedge v_j$ 可以直接写成，$v_i \vee v_j$ 可写成 $v_i + v_j$，不过必须注意这是逻辑运算，必须服从逻辑运算的法则。

故 $$\bar{\varphi} = (v_1 + \bar{v}_2)(\bar{v}_{\ } + \bar{v}_3)(\bar{v}_{\ } + \bar{v}_4)(\bar{v}_2 + \bar{v}_4)$$
$$(\bar{v}_3 + \bar{v}_4)(v_4 + v_5)(v_4 + v_6)(v_5 + v_6)$$

但是

$$(v_{\ } + v_2)(v_{\ } + \bar{v}_3) - \bar{v}_1 \bar{v}_{\ } + v_1 v_3 + \bar{v}_{\ } v_2 + v_2 v_3$$

$$= \bar{v}_1 + \bar{v}_1\bar{v}_3 + \bar{v}_1\bar{v}_2 + \bar{v}_2\bar{v}_3 = \bar{v}_1 + \bar{v}_2\bar{v}_3。$$

$$(v_1 + \bar{v}_4)(\bar{v}_2 + \bar{v}_4) = \bar{v}_4 + v_1\bar{v}_2。$$

$$(\bar{v}_3 + v_4)(\bar{v}_4 + \bar{v}_5) = v_4 + \bar{v}_3 v_5。$$

$$(v_4 + \bar{v}_6)(v_5 + v_6) = \bar{v}_6 + \bar{v}_4\bar{v}_5。$$

所以

$$\varphi = (v + v_2 v_3)(\bar{v}_4 + v_1 v_2)(v_4 + \bar{v}_3\bar{v}_5)(v_6 + \bar{v}_4\bar{v}_5)。$$

又有：

$$(\bar{v}_4 + v_1\bar{v}_2)(\bar{v}_4 + \bar{v}_3\bar{v}_5) = \bar{v}_4 + v_1 v_2 v_4 + v_3 v_4 v_5 + v_1 v_2 v_3 v_5。$$

$$(\bar{v}_1 + v_2 v_3)(\bar{v}_6 + \bar{v}_4\bar{v}_5) = \bar{v}_1\bar{v}_6 + v_1 v_4 v_5 + \bar{v}_2 v_3 v_6 + v_2 v_3 v_4 \bar{v}_5。$$

所以

$$\begin{aligned}
\varphi = &(v_4 + \bar{v}_1 v_2 v_4 + v_3\bar{v}_4\bar{v}_5 + v_1 v_2\bar{v}_3\bar{v}_5)\cdot(\bar{v}_1\bar{v}_6 + v_1\bar{v}_4\bar{v}_5 + \bar{v}_2\bar{v}_3\bar{v}_6 + \bar{v}_2\bar{v}_3\bar{v}_4\bar{v}_5) \\
= &\bar{v}_1\bar{v}_4 v_6 + v_1\bar{v}_4\bar{v}_5 + v_2 v_3\bar{v}_4\bar{v}_6 + \bar{v}_2 v_3 v_4 v_5 + \bar{v}_1\bar{v}_2\bar{v}_4 v_6 \\
&+ \bar{v}_1\bar{v}_2\bar{v}_4 v_5 + v_1 v_2\bar{v}_3\bar{v}_4 v_6 + v_1 v_2\bar{v}_3\bar{v}_4\bar{v}_5 + v_1 v_3 v_4 v_5\bar{v}_6 \\
&+ v_1 v_3\bar{v}_4\bar{v}_5 + \bar{v}_2\bar{v}_3 v_4\bar{v}_5\bar{v}_5 + \bar{v}_2\bar{v}_3 v_4 v_5 + \bar{v}_1\bar{v}_2 v_3 v_5 v_6 \\
&+ v_1 v_2 v_3 v_4 v_5 + v_1\bar{v}_2 v_3 v_5 v_6 + v_1\bar{v}_2\bar{v}_3\bar{v}_4 v_5 \\
= &v_1 v_4 v_6 + \bar{v}_1\bar{v}_4\bar{v}_5 + \bar{v}_2\bar{v}_3\bar{v}_4\bar{v}_4 + v_2 v_3 v_4\bar{v}_5 + v_1 v_2 v_3 v_5 v_6。
\end{aligned}$$

故得下面的诸极大独立集：

$$\{v_2, v_3, v_5\}, \{v_2, v_3, v_6\}, \{v_1, v_5\}, \{v_1, v_6\}, \{v_4\}。$$

§8 支 配 集

对于图$G=(V,E)$，凡是满足下列条件的顶点集合叫做图G的支配集，即G的任何顶点或属于该集合，或至少与该集合的一个顶点相邻。例图 7-4-4 中$\{v_1, v_2, v_3, v_4\}$是一支配集，$\{v_1, v_4, v_5\}$也是一个支配集，$\{v_4\}$也是一支配集。

支配集也有其实际背景，如要在$v_1, v_2, \cdots, v_n$这n座城市中建起一个通讯系统，为此，要从这n座城镇中选出几座来，在那里建立起通讯站，使得它能和所有城市相邻，这就导致找数目最少的支配集。还可以在这n个城镇中建立起两套通讯系统，使得当一个系统出了故障时，另一套系统可以取而代之而不至于陷于瘫痪。要求这两套通讯系统不在同一个地方建立通讯站，这是可能的。例如图 7-4-4 中$\{v_4\}$，$\{v_1, v_5\}$，$\{v_1, v_6\}$都是支配集，然而$\{v_4\}$与$\{v_1, v_6\}$的交集为空。

图$G=(V,E)$的某顶点集合，若满足下列两个条件时称为图G的极小支配集。这两个条件是：

(1) 本身是支配集；

(2) 从该集合中任意取走一个顶点时便破坏它作为图G的支配集。

显然有：

(1) 完全图的任意顶点都是一极小支配集；

(2) 任一支配集必包含一极小支配集；

(3) 图 G 可以有若干个极小支配集，其所含顶点数目可能不相等；

(4) 极大独立集必定是极小支配集，反之不一定成立。

(5) 图 G 的极小支配集中点数最少的数目用 $\alpha(G)$ 表示，显然有：

$$\alpha(G) \leqslant \beta(G)。$$

下面给出找所有极小支配集的方法。

对应于图 $G=(V,E)$ 的每一个顶点 $v_i \in V(G)$，有一布尔变量 v_i，以及布尔表达式：

$$\varphi_i = v_i + \sum_{v_j \in Adj(v_i)} v_j$$

作

$$\psi = \varphi_1\varphi_2\cdots\varphi_n = \prod_{v_i \in V} \varphi_i。$$

仍以图 7-7-4 为例：

$$\varphi_1 = v_1 + v_2 + v_3 + v_4,$$
$$\varphi_2 = v_1 + v_2 + v_4,$$
$$\varphi_3 = v_1 + v_3 + v_4,$$
$$\varphi_4 = v_1 + v_2 + v_3 + v_4 + v_5 + v_6,$$
$$\varphi_5 = v_4 + v_5 + v_6,$$
$$\varphi_5 = v_4 + v_5 + v_6。$$
$$\psi = (v_1+v_2+v_3+v_4)(v_1+v_2+v_4)(v_1+v_3+v_4)$$
$$(v_1+v_2+v_3+v_4+v_5+v_6)(v_4+v_5+v_6)(v_4+v_5+v_5)。$$

上面表达式中"+"为逻辑和，积为逻辑积，故服从逻辑运算法则：

$$v_i + v_i = v_i, \qquad v_iv_i = v_i;$$
$$v_i(v_i + v_j) = v_i, \qquad v_i + v_iv_j = v_i。$$

所以

$$(v_1+v_2+v_3+v_4)(v_1+v_2+v_3+v_4+v_5+v_6) = v_1+v_2+v_3+v_4,$$
$$(v_1+v_2+v_4)(v_1+v_3+v_4) = v_1+v_2v_3+v_4,$$
$$(v_4+v_5+v_6)(v_4+v_5+v_6) = v_4+v_5+v_6。$$
$$\psi = (v_1+v_2v_3+v_4)(v_1+v_2+v_3+v_4)(v_4+v_5+v_6)$$
$$= (v_1+v_4+v_2v_3)(v_4+v_5+v_6)$$
$$= v_1v_5+v_1v_6+v_4+v_2v_3v_5+v_2v_3v_6。$$

故得极小支配集如下：

$$\{v_1,v_5\},\{v_1,v_6\},\{v_4\},\{v_2,v_3,v_5\},\{v_2,v_3,v_6\}。$$

§9 色数的一种求法

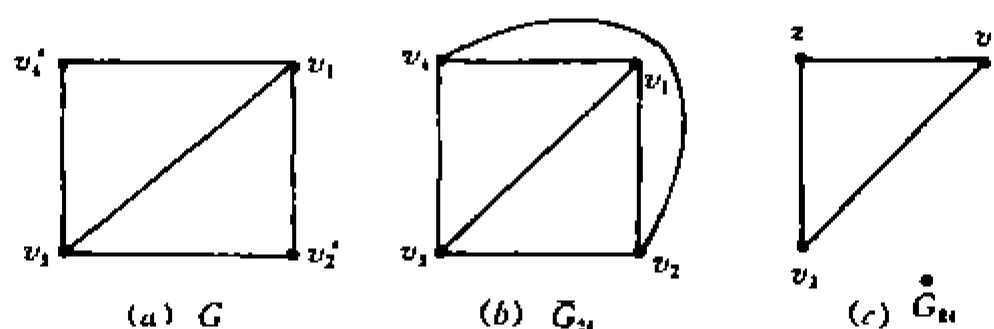
设 $G=(V,E)$，v_i，v_j 是属于 G 的两个不相邻顶点，现引进如下两个符号：

(1) $\overline{G}_{ij}$ 表示在图 G 中加上一条连 v_i，v_j 点的边所得的图；

(2) $\dot{G}_{ij}$ 表示把 v_i，v_j 两点缩为一点 z 所得的图。即图 G 中凡是与 v_i，v_j 关联的边都改

为与 z 关联。

例：如图 7 9-1 中(a)为图 G，则(b)为图 $\bar{G}_{24}$，(c)为图 $\dot{G}_{24}$。



(a) G　　(b) $\bar{G}_{24}$　　(c) $\dot{G}_{24}$

图 7 9 1

定理　设 v_i, v_j 是图 G 的两个不相邻的顶点，则图 G 的色数 $\chi(G)$ 为：

$$\chi(G)=\min\{\chi(\bar{G}_{ij}),\chi(\dot{G}_{ij})\}。$$

证明　只要证下面两个不等式

$$\chi(G)\geqslant\min\{\chi(\bar{G}_{ij}),\ \chi(\dot{G}_{ij})\},$$

$$\chi(G)\leqslant\min\{\chi(\bar{G}_{ij}),\chi(\dot{G}_{ij})\}。$$

分别成立就可以了。

先证第一个不等式。设 $k=\chi(G)$，故可用 k 种颜色使图 G 的诸相邻顶点着不同的颜色。对 v_i, v_j 两点其结果不外乎两种可能：同色或异色。若 v_i, v_j 不同色，则 k 色足以使图 $\bar{G}_{ij}$ 的相邻顶点有不同的颜色，故 $\chi(G)=\chi(\bar{G}_{ij})$。

若 v_i, v_j 颜色相同，则 k 种颜色足以使图 $\dot{G}_{ij}$ 的相邻顶点有不同颜色，故 $\chi(G)\geqslant\chi(\dot{G}_{ij})$。

由上可知，$\chi(G)$ 至少比 $\chi(\bar{G}_{ij})$、$\chi(\dot{G}_{ij})$ 两者中之一大，故比两者中最小的大。即

$$\chi(G)\geqslant\min\{\chi(\bar{G}_{ij}),\chi(\dot{G}_{ij})\}。$$

另一方面，显然有：

$$\chi(G)\leqslant\chi(\bar{G}_{ij}),$$

$$\chi(G)\leqslant\chi(\dot{G}_{ij}),$$

所以

$$\chi(G)\leqslant\min\{\chi(\bar{G}_{ij}),\ \chi(\dot{G}_{ij})\}。$$

定理证毕。

上面定理给出求图 G 色数的一种方法：

设 v_i, v_j 是图 G 中不相邻的两个顶点。v_i, v_j 有相同颜色的 G 的着色给出 $\dot{G}_{ij}$ 的一个着色；v_i, v_j 有不同颜色的 G 的着色给出 $\bar{G}_{ij}$ 的一个着色。将两种运算重复进行，直到所得到的图是完全图为止。若所得到的完全图中最小的是 K_r，那么：

$$\chi(G)=r。$$

例如：如图 7 9 2 所示：

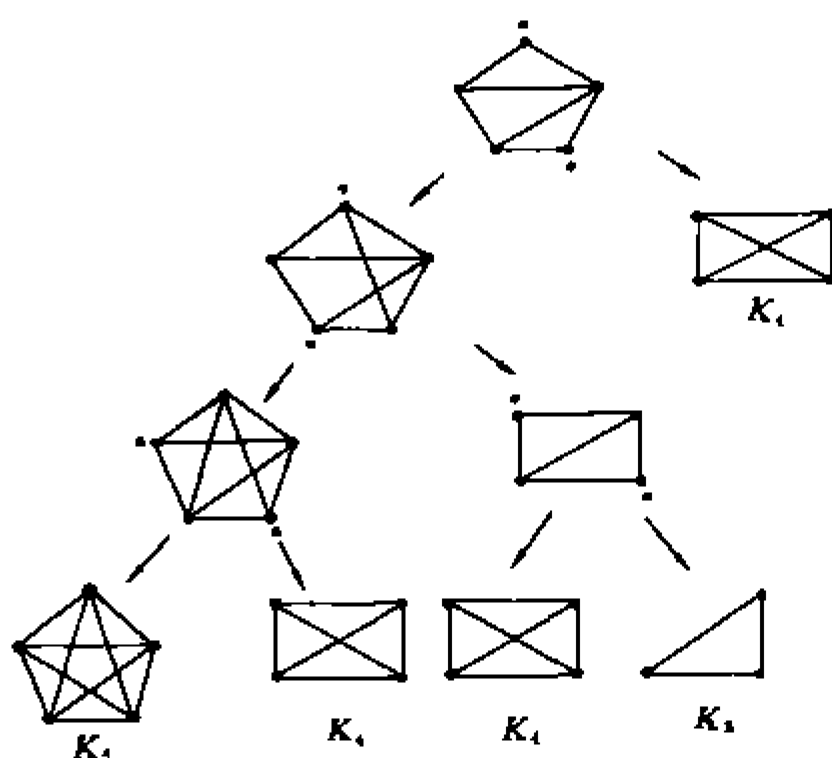


图 7-9-2

所以 $$\chi(G)=3。$$

例如可用红、黄、兰三色着色如图 7-9-3 所示。

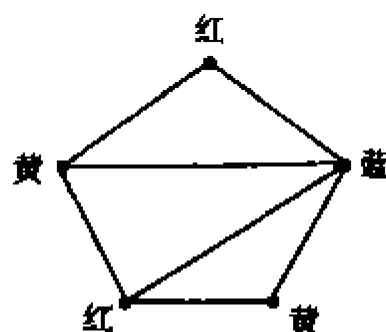


图 7-9-3

§10 色多项式

我们知道，图 G 的色数是对 G 的顶点进行着色所需要的最少的颜色数目。如果颜色的数目比色数大，自然可以对顶点着色，而且着色的方法也不止一种。现在利用色多项式来研究这个问题。

定义 设图 G 有 n 个顶点，λ 的多项式 $P(\lambda)$ 的值给出颜色数目不超过 λ 时图 G 的顶点着色不同方案数，称 $P(\lambda)$ 为图 G 的色多项式。

令 m_i 为用 i 种颜色对图 G 的顶点进行着色的不同方案数，则用 $\lambda(\geqslant i)$ 种颜色对图 G 进行着色，每次取 i 种颜色时，共有 $m_i\binom{\lambda}{i}$ 种不同着色方案，其中 $\binom{\lambda}{i}$ 为从 λ 种颜色中取 i 种的组合数。则有：

$$P(\lambda) = m_1\binom{\lambda}{1} + m_2\binom{\lambda}{2} + \cdots + m_n\binom{\lambda}{n}。$$

则

$$P(\lambda)= m_1\lambda + \frac{m_2}{2!}\lambda(\lambda - 1) + \frac{m_3}{3!}\lambda(\lambda - 1)(\lambda - 2) + \cdots$$

$$+ \frac{m_n}{n!}\lambda(\lambda - 1)(\lambda - 2)\cdots(\lambda - n + 1)。$$

故 $P(\lambda)$ 为 λ 的 n 次多项式。

两种特殊情况加以说明是必要的。

(1) 对于 n 个顶点的完全图 G 有：

$$P(\lambda) = \lambda(\lambda - 1)(\lambda - 2)\cdots(\lambda - n + 1)。$$

因第一个顶点有 λ 种选择方案，第二个顶点只有 $\lambda - 1$ 种选择方案，如此等等，依次类推，可得第 n 个顶点有 $\lambda - n+1$ 种选择方案，故

$$P(\lambda) = \lambda(\lambda - 1)(\lambda - 2)\cdots(\lambda - n + 1)。$$

(2) 若图 G 是 n 个顶点的树，则

$$P(\lambda) = \lambda(\lambda - 1)^{n-1}。$$

这个结果是显然的。

定理 若 v_i,v_j 是图 G 的不相邻二顶点，而 $P_1(\lambda)$ 是图 $\bar{G}_{ij}$ 的颜色多项式，$P_2(\lambda)$ 是图 $\dot{G}_{ij}$ 的色多项式，则图 G 的色多项式为：

$$P(\lambda) = P_1(\lambda) + P_2(\lambda)。$$

证明 用 λ 种颜色对 G 的顶点进行着色，其结果可分为两类，一是 v_i,v_j 两顶点不同色，一是 v_i,v_j 点同色。前一种有 $P_1(\lambda)$ 种方案，后一种有 $P_2(\lambda)$ 种方案，即使 v_i,v_j 二顶点有不同颜色的着色方案数等于图 $\bar{G}_{ij}$ 的着色方案数，使 v_i,v_j 二顶点有相同颜色的方案数等于图 $\dot{G}_{ij}$ 的着色方案数。所以

$$P(\lambda) = P_1(\lambda) + P_2(\lambda)。$$

定理证毕。

例如：如图 7-9-2 所示的图 G，"树叶"上的图都是完全图，其中 5 阶完全图 1 个，4 阶完全图 3 个，3 阶完全图 3 个。根据 n 个顶点的完全图的色多项式为：

$$\lambda(\lambda - 1)(\lambda - 2)\cdots(\lambda - n + 1)$$

所以

$$\begin{aligned} P(\lambda) &= \lambda(\lambda - 1)(\lambda - 2)(\lambda - 3)(\lambda - 4) + 3\lambda(\lambda - 1)(\lambda - 2)(\lambda - 3) \\ &\quad + \lambda(\lambda - 1)(\lambda - 2) \\ &= \lambda(\lambda - 1)(\lambda - 2)(\lambda^2 - 4\lambda + 4)。\end{aligned}$$

§11 色数问题应用举例

图 7-11-1 是一电路图，如要设计一印刷电路板，印刷电路板是在绝缘板内嵌入导

线，故在同一板内不允许两根导线在接点以外的地方交叉，问最少要分成几层印刷电路板?

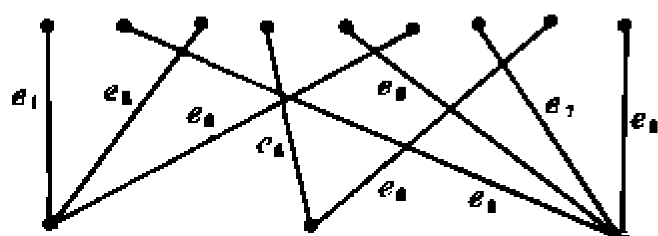


图 7-11-1

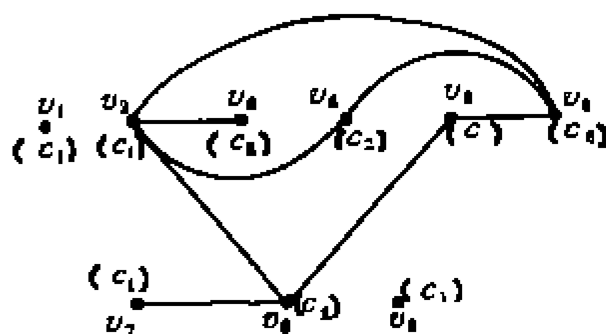
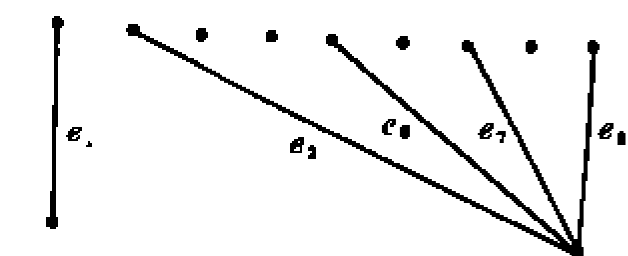


图 7-11-2

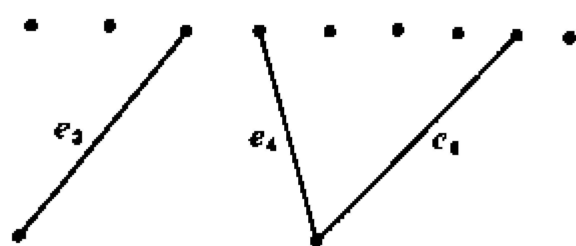
图 7-11-1 的边 $e_1,e_2,\cdots,e_9$ 分别对应于顶点 $v_1,v_2,\cdots,v_9$，当 e_i,e_j 两条边在接点以外交叉时，则过 v_i,v_j 点引一条边，得图 7-11-2。该图的色数为 3。如图 7-11-2 所示可用 c_1,c_2,c_3 三色对顶点进行着色。

由于 v_1,v_2,v_5,v_7,v_9 着同一色 c_1，而 v_3,v_4,v_8 着以 c_2 色，v_6 点着 c_3 色。故 e_1,e_2,e_5,e_7,e_9 在同一层印刷电路板上，如图 7-11-3 所示，e_3,e_4,e_8 在另一层印刷电路板上，如图 7-11-3(*b*)所示，e_6 在第三层印刷电路板上，如图 7-11-3(*c*)所示。

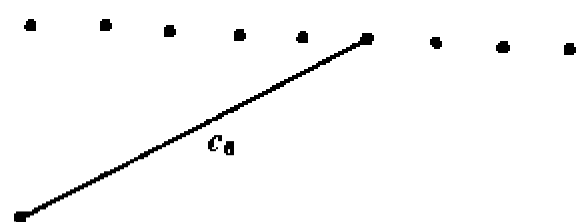
三层印刷电路板组成电路如图 7-11-4 所示。



(*a*)



(*b*)



(*c*)

图 7-11-3

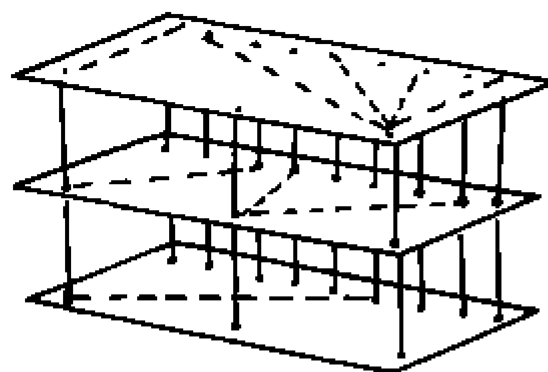


图 7-11-4

§12 PERT 图法

PERT 是 Program Evaluation Review Technique 的缩写，意即“计划审核技术”。50 年代美国一项尖端武器研制工作，参加的单位数以千计。为了安排并协调研究工作的进行，提出的一种技术，结果使得该任务比原计划大大地提前完成了。从此 PERT 图法成了安排并领导许多科研和生产任务一种的有效的方法。与 PERT 图几乎同时提出的有杜邦公司的关键路径法，也称 CPM 法。CPM 是 Critical Path Method 的缩写。CPM 法和 PERT 法是基本相同的，可称是异途同归了。

一项大的或比较大的任务，总可以按其完成工作的顺序分解成若干作业，并用一有向图来表示。这个图便是 PERT 图。图中每一条有向边表达一种作业，边上的权是完成该作业所需的时间。图中的顶点表示工作的阶段，如图 7-12-1 所示，其中共有 11 项作业组成的。

PERT 图只有一个入度为零的顶点 z，如图 7-12-1 中的“1”点，表示任务开始阶段；一个出度为 0 的顶点 $\bar{z}$，即标志着任务的结束，如图 7-12-1 中的“9”点。开始和结束是任务的两个特殊阶段。从图 7-12-1 来看，到达“5”点有两条道路，一是通过 a_1 和 a_4，总长度为 $6+1=7$；一是通过 a_2 和 a_5，总长度为 $4+1=5$。“5”标志着作业 a_1 和 a_4，a_2 和 a_5 全部完成后，才开始 a_7 和 a_8 作业的阶段。“5”这个阶段开始的时间必须在作业 a_4 和 a_5 都完成之后，即任务开始后 7 个单位时间以后，从图上来看即求从“1”点到“5”点的最长路径。

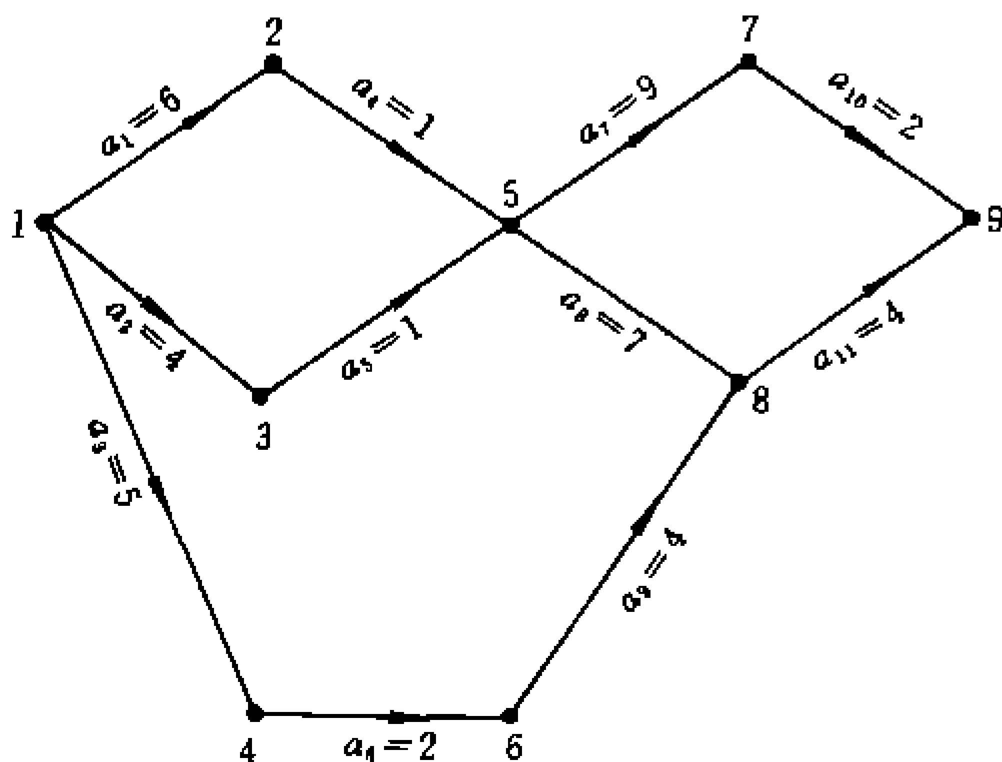


图 7-12-1

指导一项大的任务要做到胸中有数，第一个数便是“每一阶段的最早开始的时间”，即求从开始点 z 到各顶点的最长路径。比如图 7-12-1 中从开始点“1”到“8”点有 3 条路径，其中以“1”→“2”→“5”→“8”路径最长，故“8”点这个阶段应在任务开始后 $14=6+1+7$ 单位后才可开始。

从始点 z 到终点 z 也有一条长度最长的路径，称为关键路径。关键路径上的作业对能否按计划完成任务，关系重大。关键路径法也因此得名，也就是抓"关键"作业的意思。要做到胸中有数，首先要弄清楚，哪些是关键？图 7-12 1 中有两条这样的路径：

"1" → "2" → "5" → "7" → "9"；

"1" → "2" → "5" → "8" → "9"。

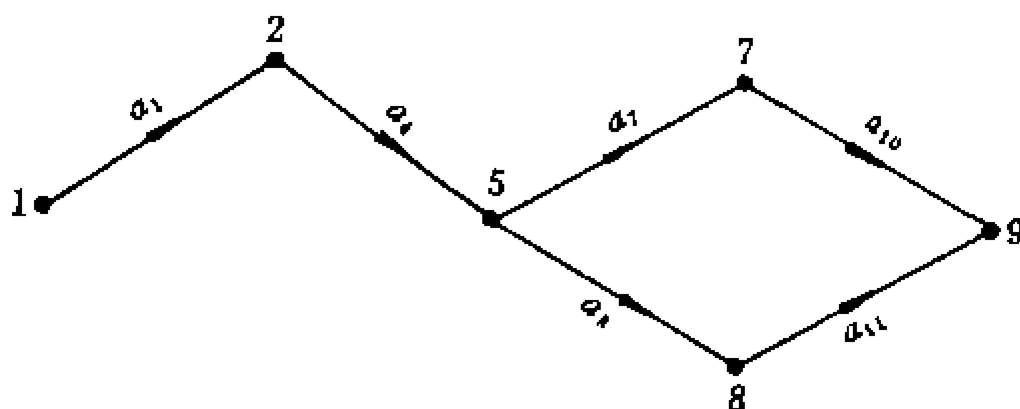


图 7-12 2

它们的长度都是 11，即该项任务的完成需要 11 个单位时间。

可以利用第三章第一节中求最短路径的方法，求从始点 z 到各点的最长路径。以图 7-12 3 为例。计算过程见图 7 12-4。最后每个顶点都有一个数，也就是各阶段开始的时间。

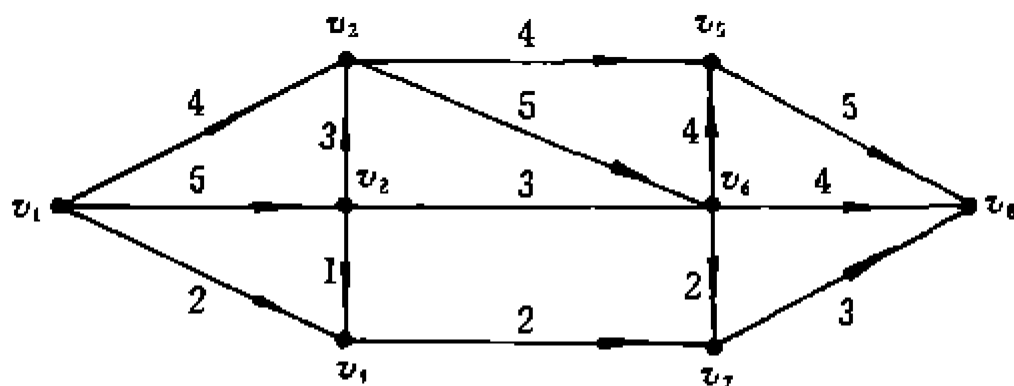


图 7 12-3

要做到"胸中有数"，还必需弄清楚各阶段最晚开始时间，再晚将影响全局。为此求各点到终点 z 的最长路径。算法（见图 7 12 5(a)），最后每一个顶点也都有一个数，用 22，这个任务完成所需的时间减去各顶点的这个数，便得各阶段最晚开始时间。图 7-12 5(b) 中各顶点有一数对，第 1 个数是最早开始时间，第 2 个数便是阶段的最晚开始时间。并不是这两个数都是一样的，比如"4"点这一对数为 (6,17)，即该阶段最早不得早于开始后的 6 单位时间；最晚也不得晚于 17 单位时间，从 6 17 这段缓冲时间为 17－6－11。同样"7"点最早开始时间为 15，最晚开始时间为 19，缓冲时间为 19 15－4。

其它各点最早开始时间和最晚开始时间相同。但仔细分析各个作业，将不难发现情况各异，"2"点和"5"点作为阶段，最早开始时间和最晚开始时间是相同的，但 (2,5) 边的长为 4，它作为作业，不一定在 8 单位时间开始，它只要保证在 17 单位时间内完成就可以了。这里 8 和 17 分别是"2"点和"5"的最晚开始时间。(2,5) 这个作业实际上只要保证不晚于 17－4－13 单位时间开始就可以了。作业的缓冲时间为 13 8－5 单位时间。

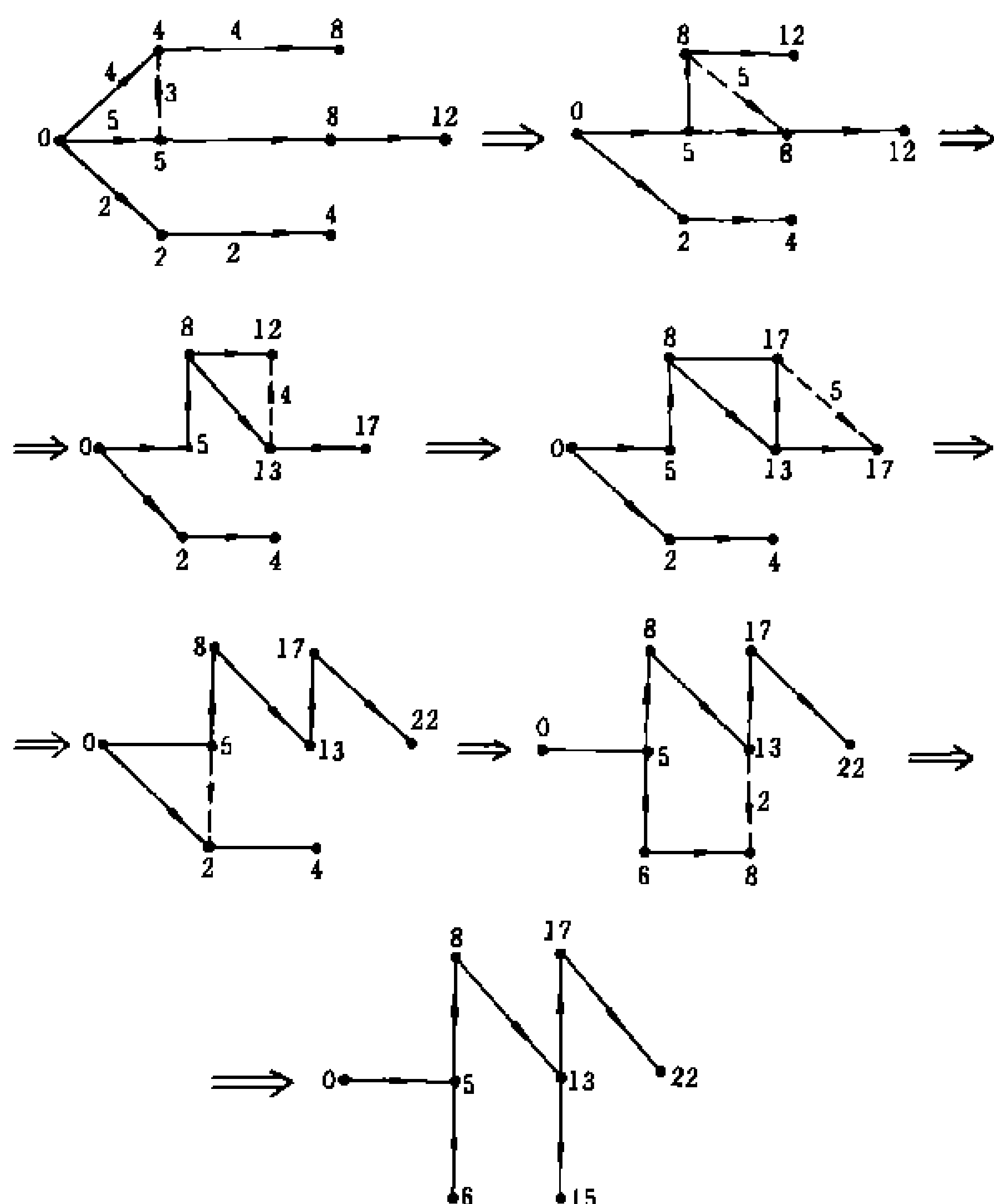


图 7-12 4

一般来说 A,B 两端点的作业 e，$t(e)$是作业完成所需的时间；$A(e_A,l_A)$表阶段 A 最早开始时间为 e_A，最晚时间为 l_A。同样理解 $B(e_B,l_B)$。

$$A(e_A,l_A) \quad t(e) \qquad B(e_B,l_B)$$

则作业 $e=(A,B)$允许缓冲的时间为

$$l_B \quad t(e)-e_A,$$

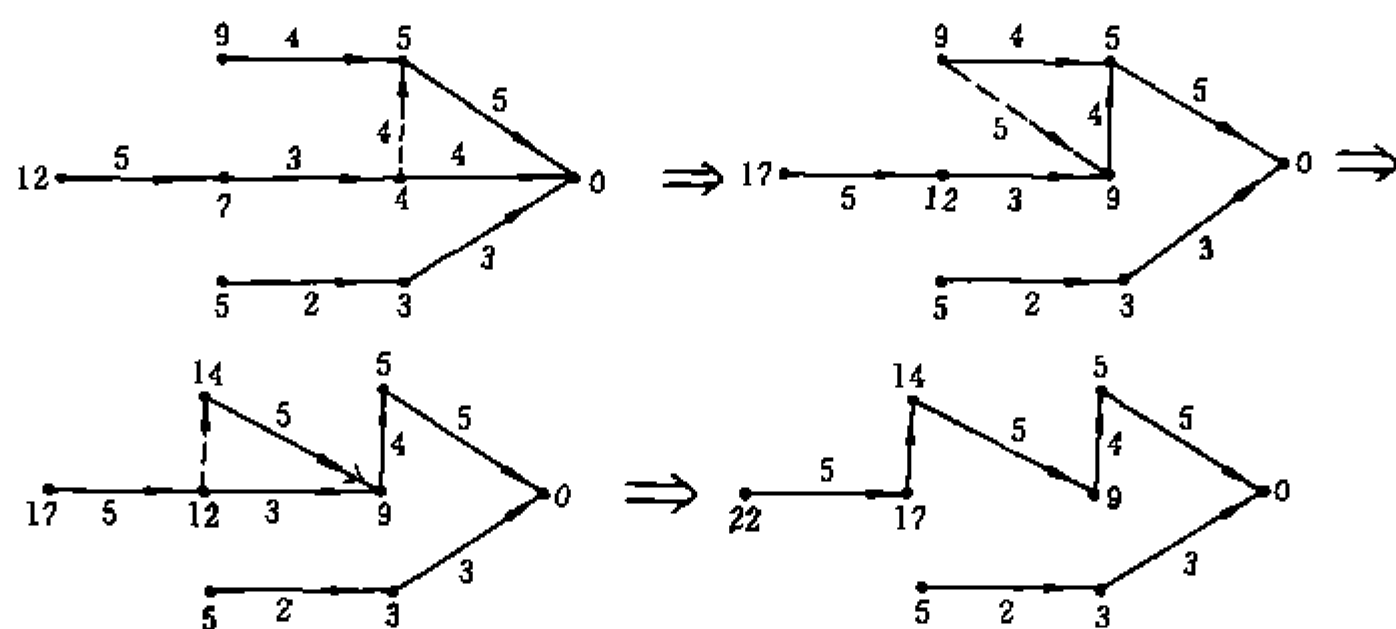
因此每个作业也有一对数偶，第 1 个数是完成所需的时间，第 2 个数是它的缓冲时间，如图 7-12 5(c)所示。

不难发现有些边它的第 2 个数是 0，即没有缓冲时间，如

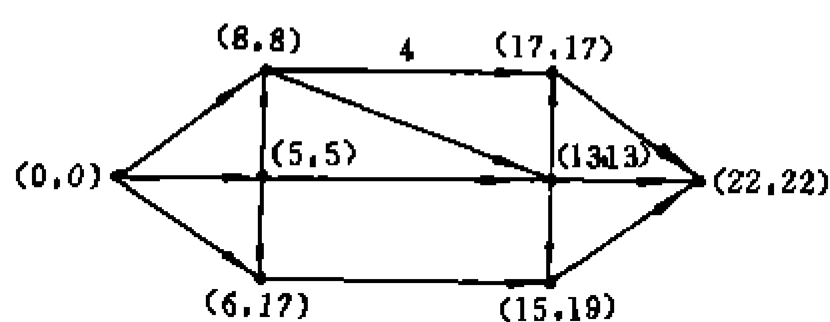
(1,3)，(3,2)，(2,6)，(6,5)，(5,8)

边。由这些边组成的从“1”点到“8”的路径，称为关键路径。

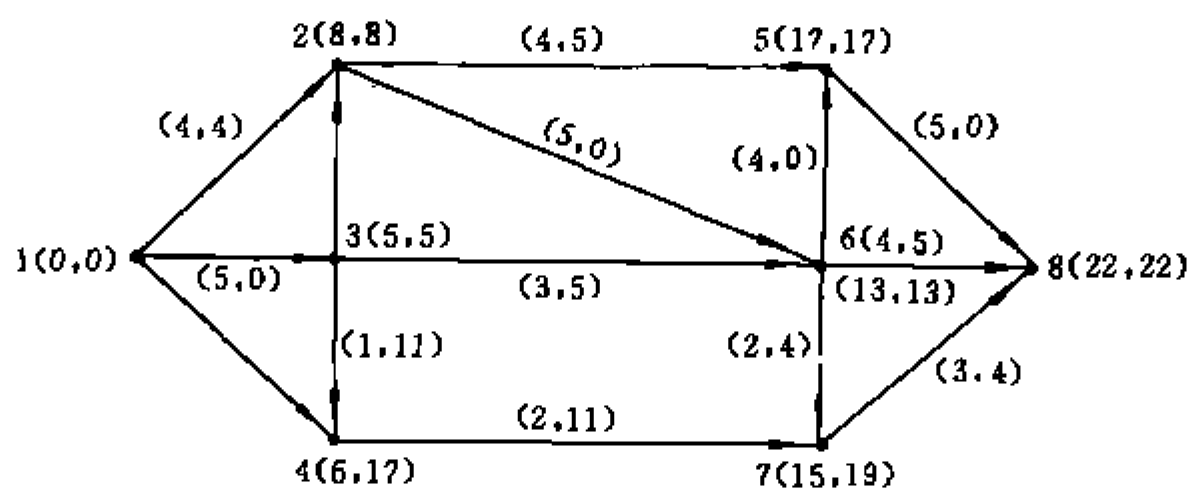
PERT 图上每个顶点，每条边都有两个数，它的意义已叙述于前，无疑它对于如何协



(a)



(b)



(c)

图 7 12-5

调各项作业的进展是有作用的。

在完成大的项目时，既要保证关键作业的按计划实现，又要避免用的人力物力，运输等过于集中在某一段时间，造成紧张局面，进行调节的只能是有缓冲时间的作业了。

§13 强连通化问题

近代化城市为了解决交通阻塞现象，采用了单行道的措施，也就是说每一条街道都给

了车辆行驶方向。但是必须做到从任何一交叉点到其任一交叉点都有道路相通。也就是可构造一城市街道交通图如下，图的顶点是街道的交叉点，每条街道是图的边。对这交通图给以方向得一有向图$G=(V,E)$。无疑这样的图必须是强连通图。下面给出一算法对图$G=(V,E)$的边给定方向，使之成为一强连通图。假定图G是连通的，而且不存在“桥”，也就是消去任何一条边都不至于破坏它的连通性。下面的算法是属于 Hopcroft-Tarjan。

(1) v 是图G的任一顶点，给v以标号，$L\leftarrow\{v\}$，$U\leftarrow V\setminus\{v\}$，$A\leftarrow\varnothing$，$l(v)\leftarrow 1$。

(2) 设w是L中l值最大的点，且v和属于U的点u相邻，$l(u)\leftarrow l(u)+1$，

$L\leftarrow L\cup\{u\}$，$U\leftarrow U\setminus\{u\}$，给$(v,u)$边以从$v$到$u$的方向，$A\leftarrow A\cup\{(v_1,u)\}$。

(3) 若$L\neq V$，重复(2)，否则转(4)。

(4) 对所有未给方向的边(x,y)以方向，若$l(x)>l(y)$，则从x指向y，结束。

算法中L是已给标号点的集合，U是未给标号的点的集合，A是已给方向的边的集合。

举例如图 7-13 1，必须说明下面的(1)表示执行算法中的步骤(1)，(2)也类似不另说明。

(1) 选v_1作为标号过程的第一个点，$l(v_1)\leftarrow 1$，$L\leftarrow\{v_1\}$，$U\leftarrow V\setminus\{v_1\}$，$A\leftarrow\varnothing$。

(2) 选$v_2\in U$，且与v_1相邻，$l(v_2)\leftarrow 2$，$L\leftarrow\{v_1,v_2\}$，$U\leftarrow\{v_1,v_1,\cdots,v_8\}$。

给(v_1,v_2)以从$v_1\to v_2$的方向，$A\leftarrow\{(v_1,v_2)\}$，

(2) $L\neq V$，返回(2)，选v_3，v_3与v_2相邻，$l(v_3)\leftarrow l(v_2)+1=3$，$L\leftarrow\{v_1,v_2,v_3\}$，$U\leftarrow\{v_4,v_5,\cdots,v_8\}$，给$(v_2,v_3)$以从$v_2\to v_3$的方向，$A\leftarrow\{(v_1,v_2),(v_2,v_3)\}$。

(2) 选v_4，$l(v_4)\leftarrow l(v_3)+1=4$，$L\leftarrow\{v_1,v_2,v_3,v_4\}$，$U\leftarrow\{v_5,v_6,\cdots,v_8\}$，给$(v_3,v_4)$边以从$v_3$到$v_4$的方向。$A=\{(v_1,v_2),(v_2,v_3),(v_3,v_4)\}$。

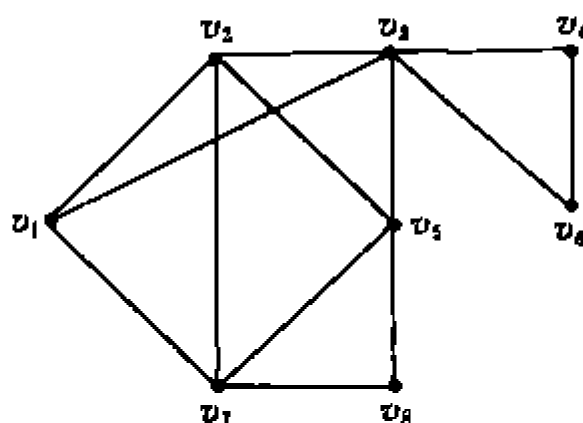


图 7 13 1

(2) 选v_6，$l(v_6)-l(v_4)+1=5$。

$L\leftarrow\{v_1,v_2,v_3,v_4,v_6\}$，$U\leftarrow\{v_5,v_7,v_8\}$。

给边(v_4,v_6)以从v_4到v_6的方向。

$$A-\{(v_1,v_2),(v_2,v_3),(v_3,v_4),(v_4,v_6)\}。$$

(2) 这时v_3是与未标号点相邻有最高标号的点，故选$v_5\in U\wedge N(v_3)$，$l(v_5)-l(v_3)+1=4$，$L=\{v_1,\cdots,v_6\}$，$U-\{v_7,v_8\}$。给(v_3,v_5)边以从v_3到v_5的方向，

$$A=\{(v_1,v_2),(v_2,v_3),(v_3,v_4),(v_4,v_6),(v_3,v_5)\}。$$

(2) 这时v_5是与未标是点相邻有最高的标号的点，故选与v_5点相邻的未标号点v_7，

$$l(v_7)=l(v_5)+1-5。L=\{v_1,\cdots,v_7\}，U-\{v_8\}。$$

给(v_5,v_7)以从v_5到v_7的方向，

$$A=\{(v_1,v_2),(v_2,v_3),(v_3,v_4),(v_4,v_5),(v_3,v_5),(v_5,v_7)\}。$$

(2) v_7 是与未标号点 v_8 点相邻标号最高的点，故选 v_8，$l(8)=l(7)+1=5$，$L=V$。$\bar{U}=\varnothing$。给边(v_7,v_8)以从 v_7 到 v_8 的方向。

$$A=\{(v_1,v_2),(v_2,v_3),(v_3,v_4),(v_4,v_6),(v_3,v_5),(v_5,v_7),(v_7,v_8)\}。$$

(4) 给未给方向的边(v_1,v_3)，(v_1,v_7)，(v_2,v_5)，(v_2,v_7)，(v_3,v_6)，(v_5,v_8)以方向，一律从标号高的点指向标号低的点。结果得图 7-13-2。

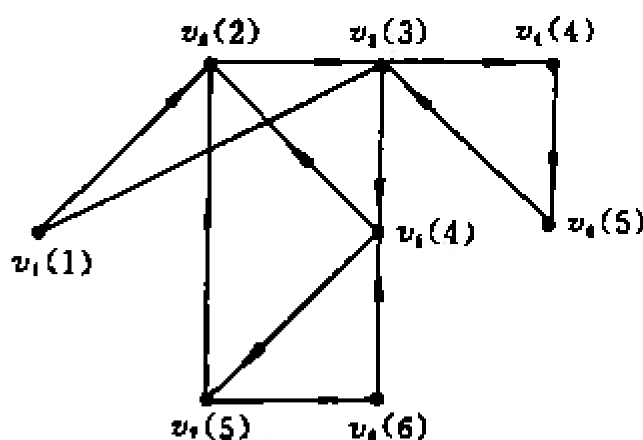


图 7-13-2

可以证明只要图 G 是没有桥的连通图，Hopcroft-Tarjan 算法总可以给出一个强连通的有向图，也就是算法的正确性。证明方法也是构造性的。证明留作作业。

习　题

1. a,b,c,d,e,f 六个人组成一小组检查五个单位的工作，若一单位和 b,c,d 三人有过工作联系，则用$\{b,c,d\}$表之。其余四个单位分别为$\{a,e,f\}$，$\{a,b,e,f\}$，$\{a,b,d,f\}$，$\{a,b,c\}$。若到一单位去检查工作的人必须和该单位没有联系的人，问应如何安排？
2. 已知用 A,B,C,D 四种材料造 Ⅰ，Ⅱ，Ⅲ，Ⅳ 四种产品的成本矩阵如下：

$$\begin{matrix} A \\ B \\ C \\ D \end{matrix}\begin{bmatrix} 99 & 6 & 59 & 73 \\ 79 & 15 & 93 & 87 \\ 67 & 93 & 13 & 81 \\ 16 & 79 & 86 & 26 \end{bmatrix}$$
$$\quad\;\; \text{Ⅰ} \quad \text{Ⅱ} \quad \text{Ⅲ} \quad \text{Ⅳ}$$

问哪种方案使成本最低？假定一种材料仅用作某种产品而不再用于其它产品。

3. $\{ace,bc,dab,db,be\}$这五个信息希望分别用它们组成字母中的一个来表示，问这是否可能？应如何安排？
4. 设 $A=(a_{ij})_{m\times n}$为布尔矩阵，$m\leqslant n$，而且 A 每行都有 k 个 1 元素，而每列 1 元素的个数不超过 k 个，则有 $A=P_1+P_2+\cdots+P_k$，其中 P_i 也是 $m\times n$ 的布尔矩阵，每行有一个 1 元素，每列的 1 元素个数不超过 1 个。
5. 求下图的色数：
6. 试证 9 个顶点 17 条边的平面图不可能用两种颜色对图的域进行染色，使得相邻的域不同色。
7. 试求下图的所有的极大独立集。

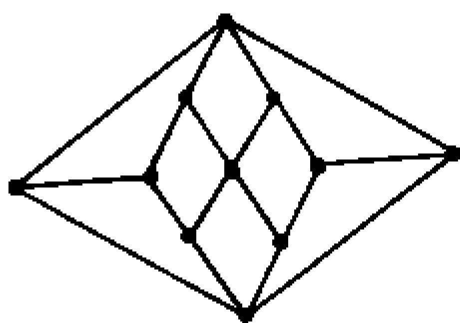


图　习题7

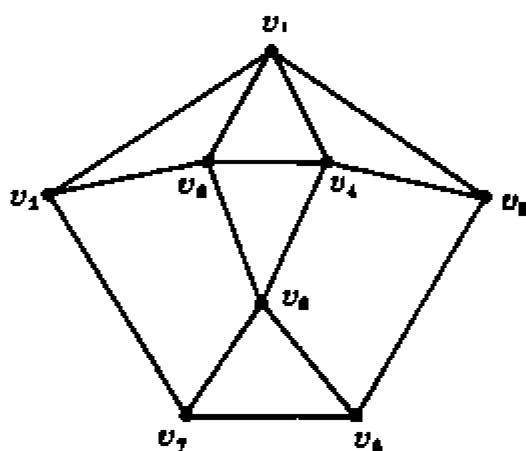


图　习题8

8. 试求上图的所有极小支配集

9. 试证 n 个顶点的树的色多项式为：

$$P(\lambda) = \lambda(\lambda - 1)^{n-1}。$$

10. 若 $P(\lambda)=\lambda(\lambda^2-6\lambda+5)Q(\lambda)$，其中 $Q(\lambda)$ 是 $n-3$ 次多项式，试证 $P(\lambda)$ 不是平面图的色多项式。

11. 现有 7 台机器，加工 7 项任务，P 为利润矩阵：

$$\boldsymbol{P} = \begin{pmatrix} 3 & 5 & 5 & 4 & 1 & 2 & 3 \\ 5 & 7 & 7 & 6 & 5 & 4 & 6 \\ 2 & 0 & 2 & 2 & 2 & 2 & 1 \\ 2 & 4 & 4 & 4 & 3 & 2 & 4 \\ 1 & 2 & 1 & 3 & 1 & 3 & 1 \\ 6 & 8 & 8 & 5 & 8 & 7 & 8 \\ 2 & 4 & 4 & 1 & 0 & 3 & 3 \end{pmatrix} = (P_{ij})$$

其中 P_{ij} 为第 i 台机器加工第 j 项任务的利润。试求最佳匹配使利润达到最大。即每台机器都加工一项任务，使利润总和达到最大。

12. 若 11 题中的矩阵是费用矩阵，问应如何匹配使费用达到最少。

13. 分别利用§6 节介绍的点着色算法，最大度优先算法，最小度最后算法对下图进行着色。

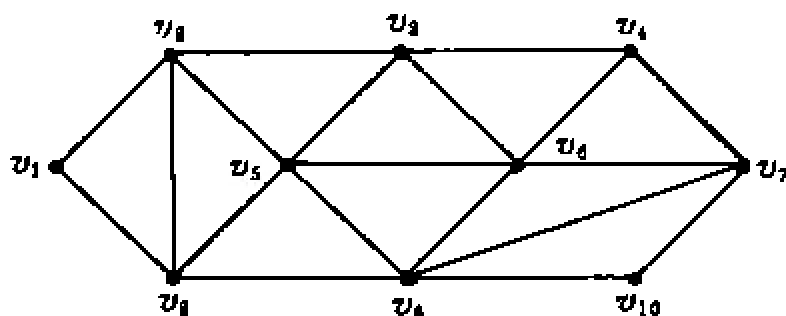
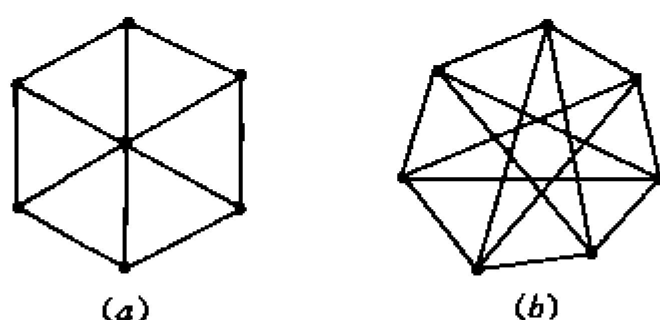


图　习题13

14. 用§6 节的三种算法对下图进行着色。

15. 利用 Hopcroft-Tarjan 算法求下图的强连通方向。



(a)　　　　　　　　(b)

图　习题 14

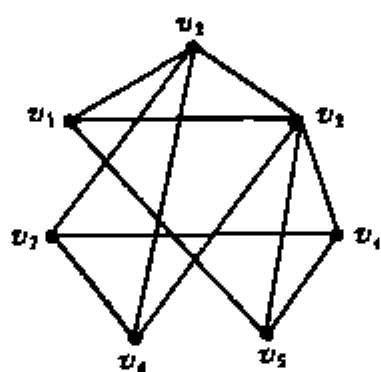


图　习题 15(a)

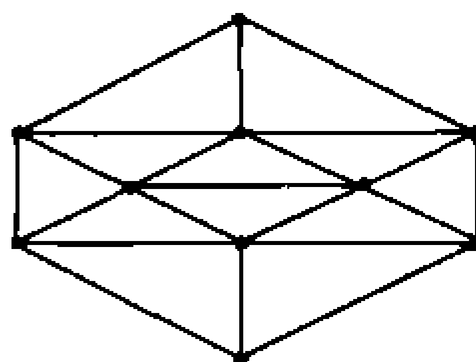


图　习题 15(b)

16. 对没有桥的连通图$G=(V,E)$应用Hopcroft Tarjan 算法得一有向图G，证下列结论成立。

(a) 当G的所有的顶点都得到标号。则集合A给出G的支撑树T。

(b) 集合A中的边不存在两端点有相同的标号l的值。

(c) (a)证明了A给出了$\bar{G}$的支撑树，试证第一个给标号的号是这个支撑树的树根，也就是说它是所有顶点的"祖先"。

(d) 设v是图$\bar{G}$上任一点，但非第一个标志点。(v,w)是A中一条边，T'是以u为"祖先"的支撑树T的子树，(v,u)不是桥，试证必存在某点$w\in T'$。w和不属于T'的点x相邻，而且边(w,x)给的方向是从w到x，x点是u的"祖先"。

(e) u是$\bar{G}$图上非支撑树根节点的任意一点则u有可达的"祖先"节点。

(f) 支撑树的根节点r是图$\bar{G}$上任一点都可到达的点。这就证明了$\bar{G}$是强连通的。